Concise User manual

MaDaM

AMS

Imprint

AMS Gesellschaft für angewandte Mess- und Systemtechnik mbH
Geschäftsführer: Dr.-Ing. Bernhard Sünder
Bahnhofstr. 6
09111 Chemnitz/ Germany

Phone.: +49 (371) 918 668 - 0
Fax: +49 (371) 918 668 - 99
E-Mail: info@AMSonline.eu
Internet: www.AMSonline.eu

Registration court: Amtsgericht Chemnitz
Registration number: HR B 9528

Copyright

Copyright © 2015 AMS GmbH
This user manual is secured by international Copyright laws.
No part of this user manual may be,
without previous written authorization by AMS GmbH,
in any form or by any means, electronic or mechanical,
including photocopying, recording or storing in an
information storage and retrieval system,
reproduced, distributed, translated or transmitt

Table of contents

1 Introduction 4
1.1 Overview 4
1.2 Features 4
2 System Overview 6
3 Technologies 7
3.1 General 7
3.2 On Server 7
3.3 On Client 7
4 Basic User Manual 8
4.1 Demo Deployment 8
4.1.1 Starting the Demo 8
4.1.2 Java Security 9
4.1.3 Terminating the Demo 9
4.2 Data model 10
4.2.1 References 10
4.2.2 Fields 10
4.2.3 Nested References 10
4.2.4 Cascading References 10
4.3 Web Frontend 11
4.3.1 Login View 11
4.3.2 Main View 11
4.3.3 Navigational Dock 12
4.3.4 Login and Language 12
4.3.5 Search Table 12
4.3.6 Facet Tree 13
4.3.7 Project Selector 13
4.3.8 Measurement Selection Table 14
4.3.9 Interaction with Measurements 14
4.3.10 Measurement Preview 15
4.3.11 Activities 16
4.3.12 Edit Data Model 16
4.3.13 Configure Jobs 18
4.3.14 Settings 19
4.4 Measurement Importer 21
4.4.1 Layout 21
4.4.2 Configuration 21
4.4.3 Importing Measurements 24
4.4.4 Validation and Uploading 25
4.4.5 Editing Measurements 27
4.4.6 Stamps 30
5 Glossary 33

1 Introduction

1.1 Overview

MaDaM is the measurement data management system developed by AMS. Management in this context means storing test data, searching stored data via web interface, and generating reports in a few simple steps.

The main focus of MaDaM's design is usability. The user interfaces are visually pleasing and easy to use. The process of importing test data is done with wizards to guide the user through this complex process, whilst allowing him to append metadata and descriptions to the raw measurement data. State of the art technologies such as Vaadin, Lucene, and JavaFX are used throughout the system.

An MDM should be a very flexible application and the possibility of extension has been strongly integrated in the design.

MaDaM currently has an extensive list of features and is improved in regular intervals.

1.2 Features

Importing Data

Instead of the outdated Java-Applet technology, the newer JavaFX Web Embedding is used. This technology is based on Java WebStart. Currently, the MaDaM Importer supports IS013499 MME, Diadem DAT and TDM files, and Excel .xslx files amongst others. Metadata related to a test is automatically extracted from its files and stored in the database. The metadata can be edited before storing the files in the database during the importing process.

Searching for Data

The stored data can be searched by using the web frontend. Searching can either be done using predefined categories or with queries. The underlying technology is called Apache Lucene and provides search capabilities with an emphasis on retrieval speed. For instance: 1000 measurements may be searched in less than 100ms.

The search query language is both intuitive and expressive and may include Boolean operators, set operators, weighting and much more. MaDaM prominently features full text search capability. Any word stored in any of the meta-data fields can be searched for.

Searching through the stored data is an activity that happens frequently, as it commonly precedes report generation. Therefore, it needs to be a highly optimized process and must offer efficient and pleasant user experience. This has been accomplished by using the Vaadin web-frontend technology, which allows searching the data without reloading the page. Everything is done in a way that is comparable to a desktop application, except the program is running in the browser. Vaadin is completely server-based and does not require any setup on the client side. Furthermore, it is very flexible in terms of extensibility and later additions.

Creating PDF Reports

There are two main methods in MaDaM that can be used for data analysis. The first way is automatic generation of PDF files according to predefined templates. The resulting PDF files can use jBEAM project files. Users can upload such project files to the server and share them with other users. Then, once a measurement is selected, either the template is applied to the data or a default layout is used, and a downloadable PDF file is generated.

Interactively Analysing Data

AMS' flagship product, jBEAM, comes bundled with MaDaM. It features report creation as well as complex analyses, including standard calculations such as FFT, order analyses, sonogram, rainflow, class limit classification, or engine characteristic fields. Furthermore, completely predefined analyses such as crash test evaluation are included. Users can store their analyses with live data or without on their personal PC or work entirely in a web context. The data can be exported to several different data formats, e.g. ASCII, DIAdem, Excel, Gidas, Matlab, ISO13499, or MDF.

LDAP Integration

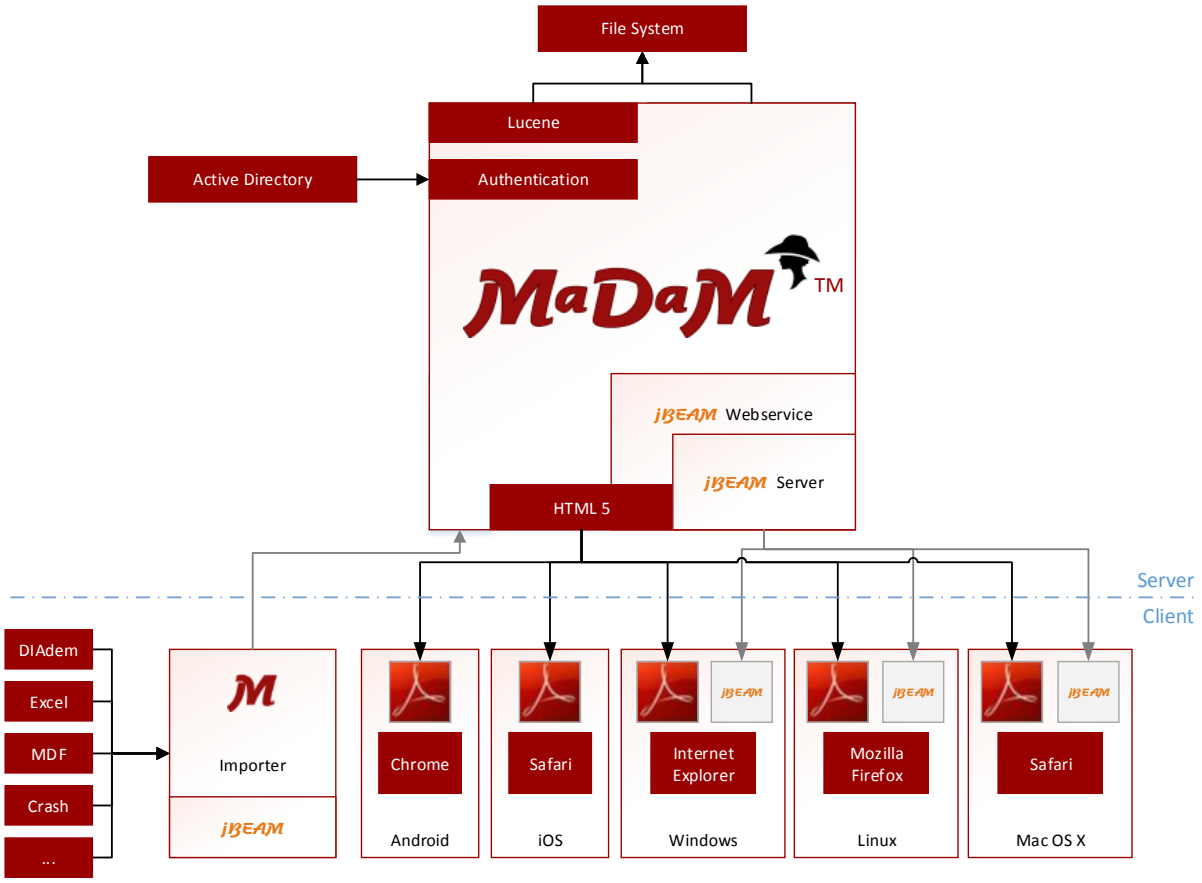
MaDaM can integrate Microsoft Windows domain servers. Hence there is no necessity to manage users separately. However, if you wish to do so, MaDaM can handle user management on its own, for example in environments where LDAP is not available.

Mobile Device Support

The MaDaM web-frontend can be accessed with desktop devices as well as mobile devices. MaDaM allows the test engineer to generate and view PDFs on-the-go. This way measurement data can be accessed directly in the testing area and stacks of printed reports become obsolete.

Currently, iOS and Android devices are supported; this includes the Apple iPad and iPad 2, or Android tablets such as the Google Nexus 10.

2 System Overview



Web-Frontend

The web frontend is the main entry point of the user. Its frontend technology is provided by the Vaadin framework library. The user navigates to the frontend with a browser of his choice. Afterwards it is possible to search the system, create reports from existing projects, or start the Importer via the embedded Java WebStart Technology.

Importer

The Importer is the component used to import measurements into the database. It is also used to manage and edit the meta-information such as laboratories or test engineers. This component communicates with the server via an EnCom endpoint. Access to the importer can be restricted, for instance to system administrators.

EnCom Endpoint

The EnCom endpoint is the main communication endpoint of the system. It uses AMS' own EnCom protocol to provide an access to the data storage and the index. This endpoint's responsibilities include ensuring data integrity as well as access control.

Index Controller

The system uses an index based database solution instead of a relational database. This index is provided by Apache's Lucene. The index controller regulates the access to the index and provides parsing mechanisms for data and search queries alike.

The index itself is created from the file-based storage system. It is automatically updated when changes to the file system occur.

3 Technologies

3.1 General

The *MaDaM* system has been developed using Java 7. Therefore Oracle's Java 7 JVM is mandatory for all the system components. OpenJDK is not supported.

3.2 On Server

The server handles the measurement data and provides the web frontend. For the first task, the only important requirement is free space on the hard disk. The web frontend runs as a web application (.war) and needs to be deployed in a Java Application Container such as Jetty or Tomcat. The demo release includes its own Jetty for ease of use.

The server can be deployed on modern Linux systems as well as Mac OS, as long as they support the aforementioned technologies.

3.3 On Client

The client browses the web frontend using a modern browser with HTML5 functionality and JavaScript. All modern browsers (IE 9+, Mozilla Firefox, Google Chrome, and Opera) are supported. JavaScript needs to be enabled in the browser.

The importer wizards are started using Java Webstart. This means, that usage of JNLP files needs to be allowed by the system's administrator. The wizards also rely on JavaFX, which means that JavaFX needs to be installed on all client PCs with import privileges. Since Java 7 SDK 7u10, JavaFX is bundled with the SDK.

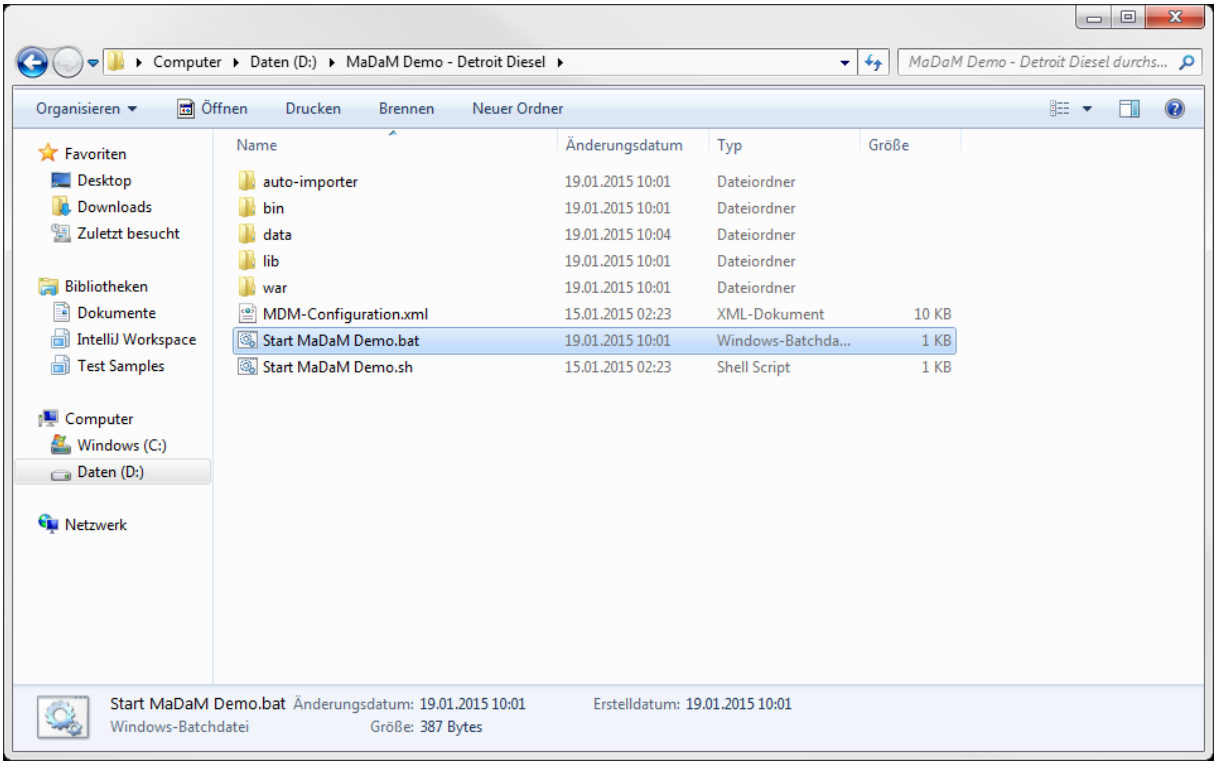
The client's operating system needs to support the aforementioned technologies, otherwise there are no constraints. Access via iOS and Android is possible, although, since Java FX is not yet available for mobile devices, the import functionality is limited to desktop PCs.

4 Basic User Manual

4.1 Demo Deployment

4.1.1 Starting the Demo

The demo system features one-click-installation for demo purposes. The system is provided in bundled form – as a zip file. On a normal operating system, just extract the zip file and open it. Then double click on 'Start MaDaM Demo.bat" batch file.



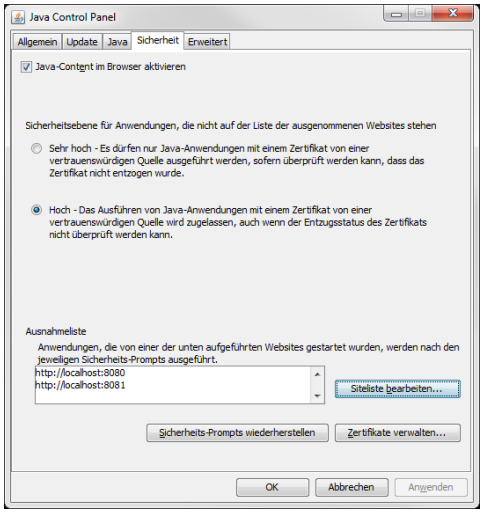
Executing this batch file opens a new window or tab of the default browser automatically navigating to the correct page. By default, this is http://localhost:8080/mdm . Should no browser window open, please navigate to this page manually.

System Requirements:

- Java 7 or newer
- 2 GB of free disk space
- 4 GB and RAM

4.1.2 Java Security

It may be necessary to change java security settings in order to correctly execute the demo.


Java Control Panel
Allgemein
Update
Java
Sicherheit
Erweitert
Java-Content im Browser aktivieren
Sicherheitsebene für Anwendungen, die nicht auf der Liste der ausgenommenen Websites stehen
Sehr hoch - Es dürfen nur Java-Anwendungen mit einem Zertifikat von einer vertrauenswürdigen Quelle ausgeführt werden, sofern überprüft werden kann, dass das Zertifikat nicht entzogen wurde.
Hoch - Das Ausführen von Java-Anwendungen mit einem Zertifikat von einer vertrauenswürdigen Quelle wird zugelassen, auch wenn der Entzugsstatus des Zertifikats nicht überprüft werden kann.
Ausnahmeliste
Anwendungen, die von einer der unten aufgeführten Websites gestartet wurden, werden nach den jeweiligen Sicherheits-Prompts ausgeführt.
http://localhost:8080
http://localhost:8081
Siteliste bearbeiten...
Sicherheits-Prompts wiederherstellen
Zertifikate verwalten...
OK
Abbrechen
Anwenden

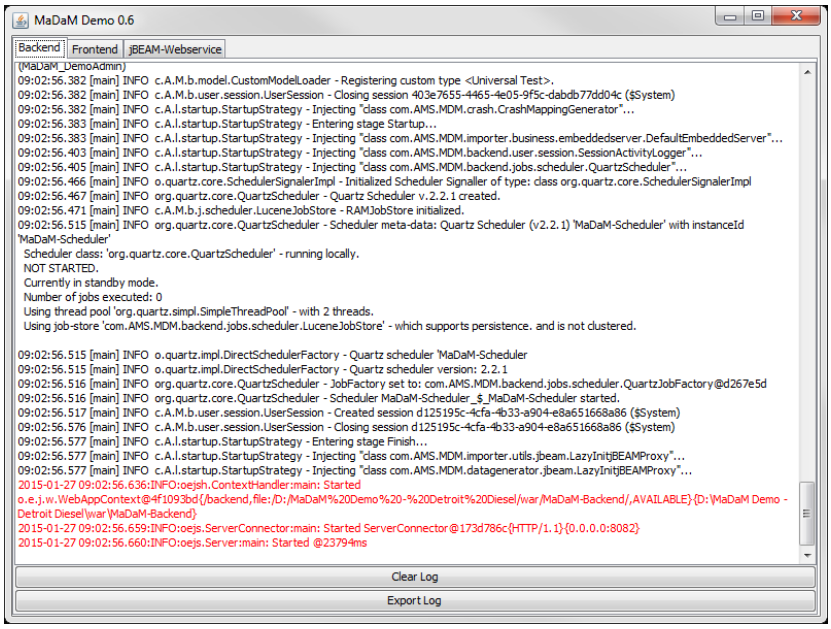
Changing the java security settings can be done in the 'Java Control Panel'. The panel can be accessed via 'Windows Start Menu' > 'Programs' > 'Java' for Windows or via 'System Preferences' > 'Java' for Mac.

After opening the panel, navigate to the 'Security' tab, and select 'Edit Sitelist'. In the following popup dialog, you need to enter the addresses used to access *MaDaM*. By default, these are 'http://localhost:8080' and 'http://localhost:8081' for the demo system.

After whitelisting these addresses, the demo can be used as intended.

4.1.3 Terminating the Demo

As the demo itself starts a server, simply closing the browser with which the user connects to the system does not suffice to terminate the demo.


MaDaM Demo 0.6
Backend
Frontend
jBEAM-Webservice
(MaDaM_DemoAdmin)
09:02:56.382 [main] INFO c.A.M.b.model.CustomModelLoader - Registering custom type <Universal Test>.
09:02:56.382 [main] INFO c.A.M.b.user.session.UserSession - Closing session 403e7655-4465-4e05-9f5c-dabdb77dd04c ($System)
09:02:56.382 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.crash.CrashMappingGenerator"...
09:02:56.383 [main] INFO c.A.l.startup.StartupStrategy - Entering stage Startup...
09:02:56.383 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.importer.business.embeddedserver.DefaultEmbeddedServer"...
09:02:56.403 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.backend.user.session.SessionActivityLogger"...
09:02:56.405 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.backend.jobs.scheduler.QuartzScheduler"...
09:02:56.466 [main] INFO o.quartz.core.SchedulerSignalerImpl - Initialized Scheduler Signaller of type: class org.quartz.core.SchedulerSignalerImpl
09:02:56.467 [main] INFO org.quartz.core.QuartzScheduler - Quartz Scheduler v.2.2.1 created.
09:02:56.471 [main] INFO c.A.M.b.j.scheduler.LuceneJobStore - RAMJobStore initialized.
09:02:56.515 [main] INFO org.quartz.core.QuartzScheduler - Scheduler meta-data: Quartz Scheduler (v2.2.1) 'MaDaM-Scheduler' with instanceId 'MaDaM-Scheduler'
Scheduler class: 'org.quartz.core.QuartzScheduler' - running locally.
NOT STARTED.
Currently in standby mode.
Number of jobs executed: 0
Using thread pool 'org.quartz.simpl.SimpleThreadPool' - with 2 threads.
Using job-store 'com.AMS.MDM.backend.jobs.scheduler.LuceneJobStore' - which supports persistence. and is not clustered.
09:02:56.515 [main] INFO o.quartz.impl.DirectSchedulerFactory - Quartz scheduler 'MaDaM-Scheduler
09:02:56.515 [main] INFO o.quartz.impl.DirectSchedulerFactory - Quartz scheduler version: 2.2.1
09:02:56.516 [main] INFO org.quartz.core.QuartzScheduler - JobFactory set to: com.AMS.MDM.backend.jobs.scheduler.QuartzJobFactory@d267e5d
09:02:56.516 [main] INFO org.quartz.core.QuartzScheduler - Scheduler MaDaM-Scheduler_$_MaDaM-Scheduler started.
09:02:56.517 [main] INFO c.A.M.b.user.session.UserSession - Created session d125195c-4cfa-4b33-a904-e8a651668a86 ($System)
09:02:56.576 [main] INFO c.A.M.b.user.session.UserSession - Closing session d125195c-4cfa-4b33-a904-e8a651668a86 ($System)
09:02:56.577 [main] INFO c.A.l.startup.StartupStrategy - Entering stage Finish...
09:02:56.577 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.importer.utils.jbeam.LazyInitjBEAMProxy"...
09:02:56.577 [main] INFO c.A.l.startup.StartupStrategy - Injecting "class com.AMS.MDM.datagenerator.jbeam.LazyInitjBEAMProxy"...
2015-01-27 09:02:56.636:INFO:oejsh.ContextHandler:main: Started
o.e.j.w.WebAppContext@4f1093bd{/backend,file:/D:/MaDaM%20Demo%20-%20Detroit%20Diesel/war/MaDaM-Backend/,AVAILABLE}{D:\MaDaM Demo - Detroit Diesel\war\MaDaM-Backend}
2015-01-27 09:02:56.659:INFO:oejs.ServerConnector:main: Started ServerConnector@173d786c{HTTP/1.1}{0.0.0.0:8082}
2015-01-27 09:02:56.660:INFO:oejs.Server:main: Started @23794ms
Clear Log
Export Log

In order to exit the demo itself, it is necessary to close the java window with the label 'MaDaM Demo 0.6'.

4.2 Data model

The *MaDaM* system stores data in a specific, predefined manner. Understand the structure of this storage is crucial for the understanding of certain parts of the handling of *MaDaM*.

4.2.1 References

In the *MaDaM* context, a reference is a data structure representing information about a specific object. It consists of a type and fields. A field is a value with a name. While the type reflects the kind of object that is described, the fields hold the meta-information about the object itself. For example, a reference describing a photo would consist of the type – 'Photo' – and a number of fields describing meta-information regarding that photo, like aspect ratio or height and width.

4.2.2 Fields

There are two different types of fields any reference can have: *model fields* and *additional fields*.

Model fields are predefined by the data model and typically need to have a value. They are meant to distinguish different data entries.

Furthermore, model fields can have various different value formats. A model field can contain characters in various formats, other references, or files. The character values can be textual information, numerical values or dates. While it is possible to store a date as text, storing it directly as a date has the advantage that it can be searched for more specifically.

Additional fields, on the other hand, are limited to textual information. They are not predefined and can thus be added ad hoc.

4.2.3 Nested References

As mentioned above, fields can contain other references – either a single reference or a list thereof. In the context of *MaDaM*, a reference that is held by a field in another reference is called a 'nested reference'. Consequently, the encompassing reference is referred to as 'parent' or 'parent reference'.

An example for nested references can be found in tests. A measurement always consists of various recorded channels. *MaDaM* reflects this, as any measurement – which is a test reference – must always contain one or more channels. Thus, the test reference is the parent, and the channels are nested references.

Generally, references can be nested in multiple parents.

4.2.4 Cascading References

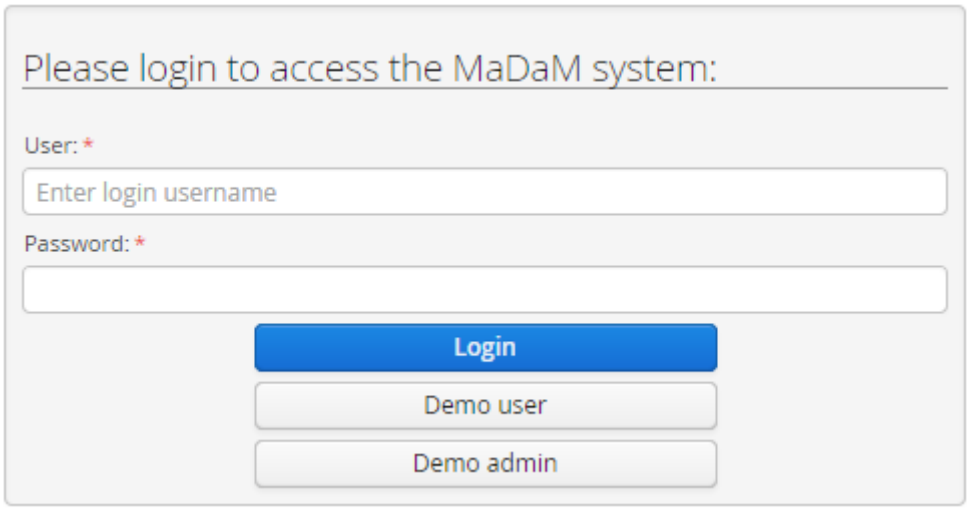
While references can be nested in multiple parents, there is an exception: cascading references. These are only nested in one parent, and anything that affects the parent (e.g. deletion) also affects the cascading references.

Therefore, cascading references occur only once in the system – in connection to their respective parent. Photos can be used as an example here. While generally tests are connected to photos, the photos are usually exclusive to one specific test. They are uploaded when the test is uploaded and need to be deleted when the test is deleted. If the photos would not be deleted with their parent – the test – they would occupy space while not being accessible.

4.3 Web Frontend

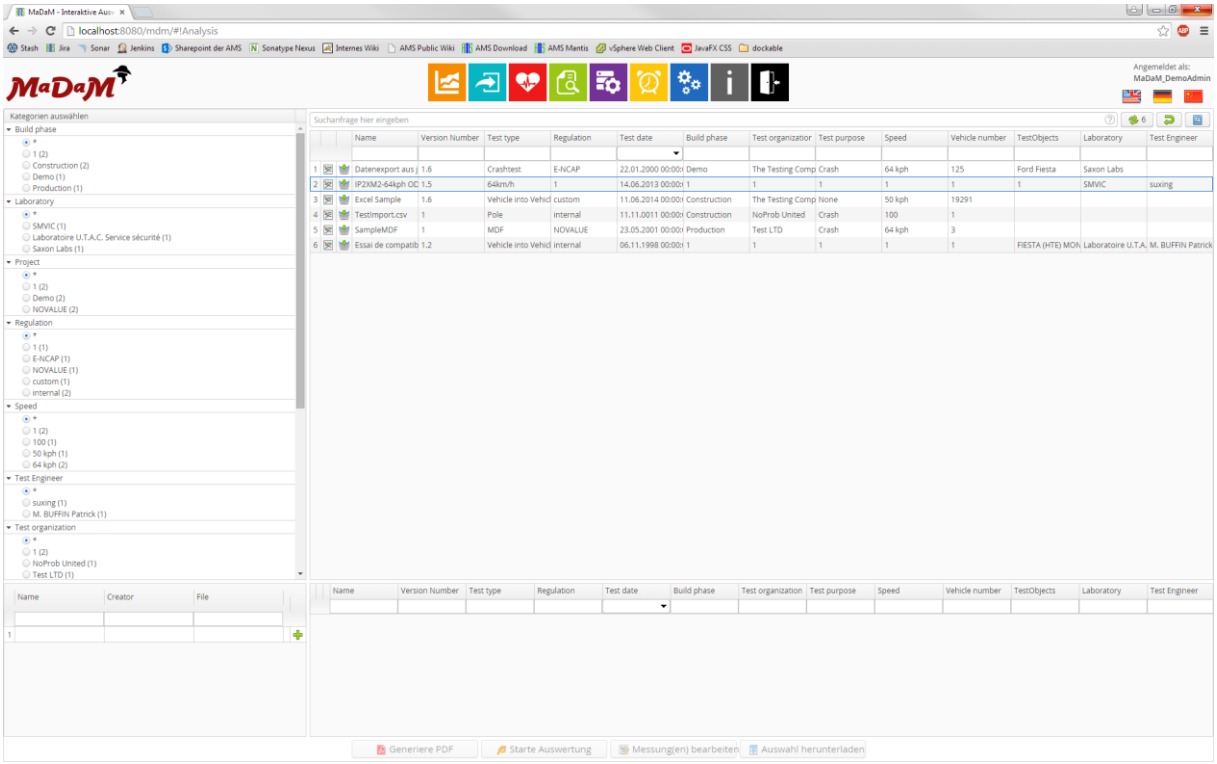
The Web Frontend is the access point for users wishing to interact with the system. It can be accessed from any HTML5 capable browser regardless of underlying operating system or device.

4.3.1 Login View



The login view is what any user first sees upon opening the system in the browser. The login process is just as on any website. For the purpose of easy demonstration, two demo-login buttons are included.

4.3.2 Main View



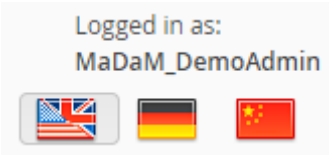
The main View is the core of the system. The user can create search for stored measurements, create reports or navigate to other components. The individual components will be covered one after the other.

4.3.3 Navigational Dock

The navigational dock is located at the top of any MaDaM web page. It is used to navigate, and the contained components from left to right are:

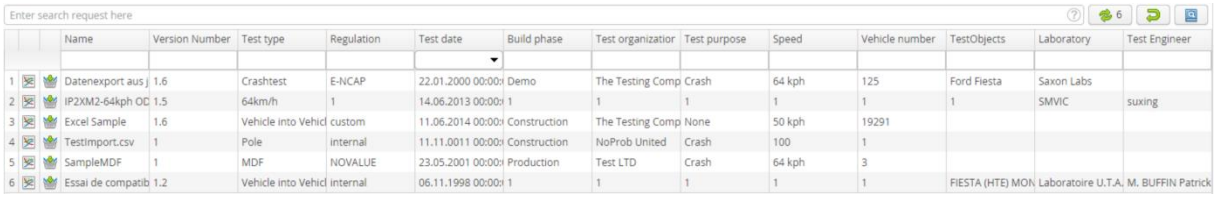
- **Create Analysis** - shows the main view of the system
- **Import Measurements** - starts the MaDaM importer to store measurements
- **Activities** - opens a log of past interactions with the system
- **Reference View** - used to navigate more specific information contained in the system
- **Edit data Model** - opens a view that can configure the underlying data structure
- **Configure jobs** - allows the user to define automated interactions with the system
- **Settings** - configuration dialog for default table settings and default access rights
- **About** - contains more information about the system itself
- **Logout** - logs the current user out

4.3.4 Login and Language



This information is found in the top-right corner of the screen. It is shown what user is currently viewing the page and what language is used. The language can be changed ad hoc.

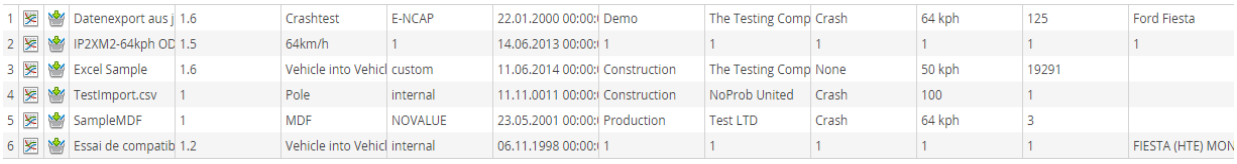
4.3.5 Search Table


Enter search request here
Name
Version Number
Test type
Regulation
Test date
Build phase
Test organizatior
Test purpose
Speed
Vehicle number
TestObjects
Laboratory
Test Engineer
1 Datenexport aus j 1.6 Crashtest E-NCAP 22.01.2000 00:00: Demo The Testing Comp Crash 64 kph 125 Ford Fiesta Saxon Labs
2 IP2XM2-64kph OD 1.5 64km/h 1 14.06.2013 00:00: 1 1 1 1 1 SMVIC suxing
3 Excel Sample 1.6 Vehicle into Vehicl custom 11.06.2014 00:00: Construction The Testing Comp None 50 kph 19291
4 TestImport.csv 1 Pole internal 11.11.0011 00:00: Construction NoProb United Crash 100 1
5 SampleMDF 1 MDF NOVALUE 23.05.2001 00:00: Production Test LTD Crash 64 kph 3
6 Essai de compatib 1.2 Vehicle into Vehicl internal 06.11.1998 00:00: 1 1 1 1 1 FIESTA (HTE) MON Laboratoire U.T.A. M. BUFFIN Patrick

The search table is located on the Create Analysis view. The user can resize or reorder the columns at all times. Furthermore, in the top right corner of the search table is a small gear button that becomes visible when the mouse comes close to it. Via this button, the user can select which columns should be displayed.

Additionally, it is possible to sort the entries by clicking on the column headers. The text fields below the column headers serve as additional filters.

The search table contains the search bar – the text field at the top – which enables users to search for measurements by utilizing queries. The query language itself consists of two parts: free text query and phrases.


1 Datenexport aus j 1.6 Crashtest E-NCAP 22.01.2000 00:00: Demo The Testing Comp Crash 64 kph 125 Ford Fiesta
2 IP2XM2-64kph OD 1.5 64km/h 1 14.06.2013 00:00: 1 1 1 1 1 1
3 Excel Sample 1.6 Vehicle into Vehicl custom 11.06.2014 00:00: Construction The Testing Comp None 50 kph 19291
4 TestImport.csv 1 Pole internal 11.11.0011 00:00: Construction NoProb United Crash 100 1
5 SampleMDF 1 MDF NOVALUE 23.05.2001 00:00: Production Test LTD Crash 64 kph 3
6 Essai de compatib 1.2 Vehicle into Vehicl internal 06.11.1998 00:00: 1 1 1 1 1 FIESTA (HTE) MON

Free text queries mean that a user can simply type in keywords and the system searches for them intelligently. In the above example, it would be possible to simply type ‘internal Construction” and the system would find the fourth measurement, as only this measurement contains ‘internal” in one field and ‘Construction” in another.

Query phrases can be much more complex. Consider the following example:

An engineer wants to search for all measurements, which have a channel starting with ‘HIC” with a maximum above 100. A query reflecting this would be ‘DataObjects.(Name=”HIC*” and Maximum>100)”. In this query, the ‘*” is a wildcard, meaning that any number of characters can be following the search term ‘HIC” for the name field.

In order to assist users with queries, there is a button with a question mark at the rightmost part of the search field, which opens a help dialog with further explanations.

4.3.6 Facet Tree

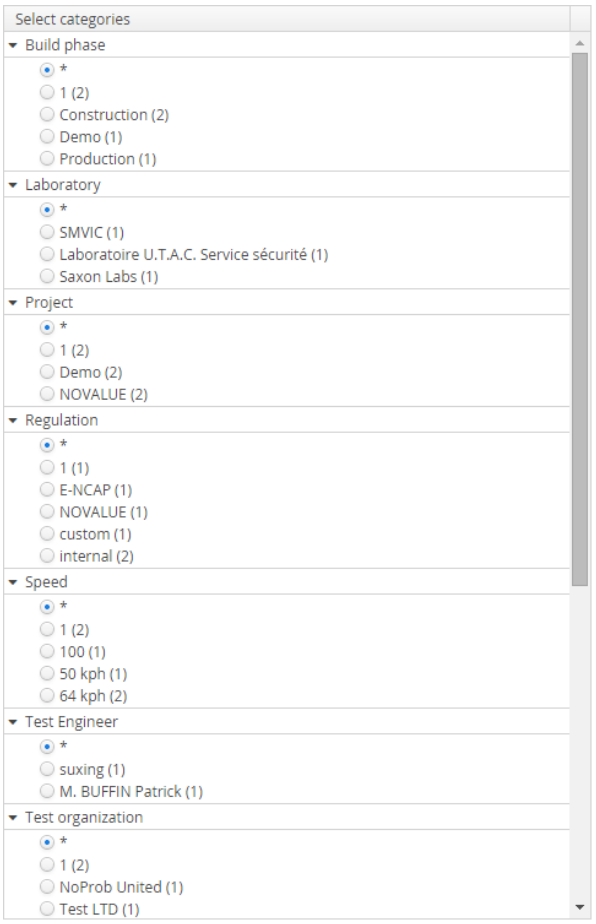
The facet tree offers selectors which can be used to filter search results. It is divided into several categories and displays every possible value for this category. The number next to each value represents the amount of stored measurements that would match this value.

By clicking a checkbox next to a value, the search result is filtered so that only results that match the value are visible.

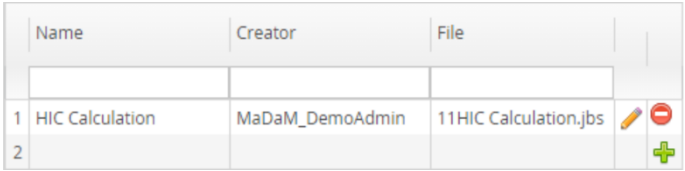
It is possible to select multiple facets in order to combine them. This would result in an intersection of the results for each individual facet.

The facet tree can also be used in concert with queries entered into the search bar.

In the depicted example, searching for all E-NCAP tests done by the test organization Test LTD with 50 kph is as easy as selecting the appropriate facets.

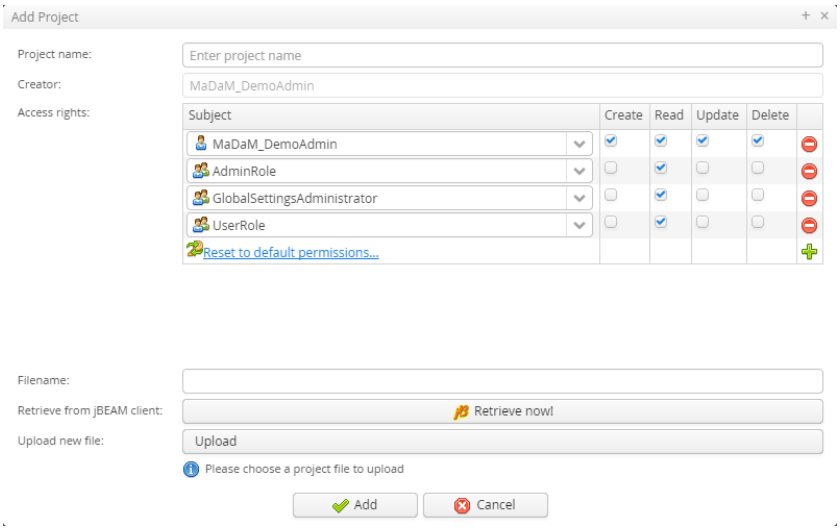


4.3.7 Project Selector



The project selector is used to upload projects to the system or to apply projects that are used for pdf creation or interactive analysis. New projects may be added, existing projects edited or deleted.

Upon either clicking the green ‘+” for adding new projects or the pencil for editing an existing one, a popup window is opened in which project information can be edited.

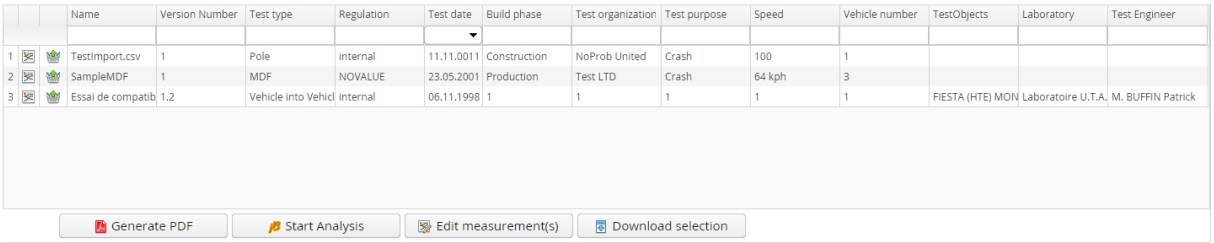


Every project needs a name, a creator and a *jBEAM* project file associated with it.

If no project name is set, a default value will be used.

Similarly, if there is no creator set, the currently logged in user will automatically be used as creator.

4.3.8 Measurement Selection Table



The measurement selection table is the area at the bottom of the main view. It contains a number of measurements and a set of buttons.

The measurements listed in this table are the ones that are used for pdf creation, interactive analysis, editing or downloading.

Measurements can be moved to and from this table with two methods. Firstly, it is possible to simply drag and drop measurements into the selection table area or from the selection table into the search table. Secondly, the user can click the basket icon with the green arrow downwards – including it in the selection table – or the basket icon with the green arrow upwards – removing it from the selection table.

4.3.9 Interaction with Measurements

The interaction with measurements included in the selection table can be done via this set of buttons.

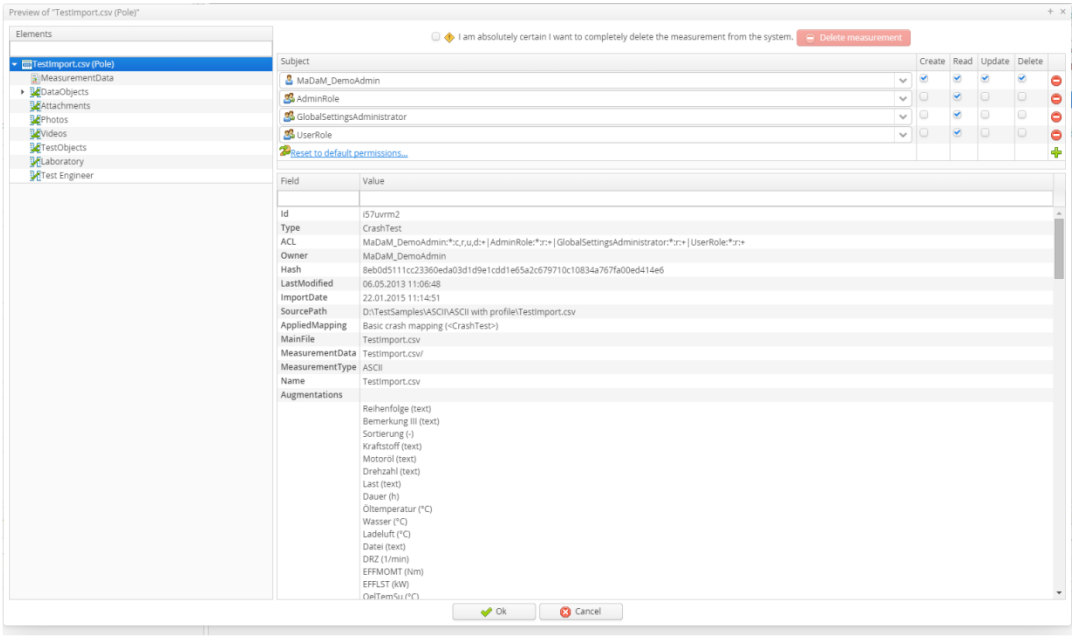
The **Generated PDF** button opens a new tab, in which a report will be generated. Either a default layout will be used, or the project selected via the Project Selector – in case this project contains visual elements.

Clicking the **Start Analysis** button starts the download of a JNLP file which is used to start *jBEAM* using Java WebStart technology. Any selected projects are applied to this analysis as well.

Edit measurement(s) starts the download of a JNLP file as well. However, this JNLP is used to start the *MaDaM* Importer, as this is the only way to edit measurements. If a user has a *MaDaM* Importer running, the download is skipped and the measurement data is sent directly to the opened Importer.

Finally, the **Download selection** button starts a zip file download. All measurements currently listed in the selection table are included in the zip file.

4.3.10 Measurement Preview

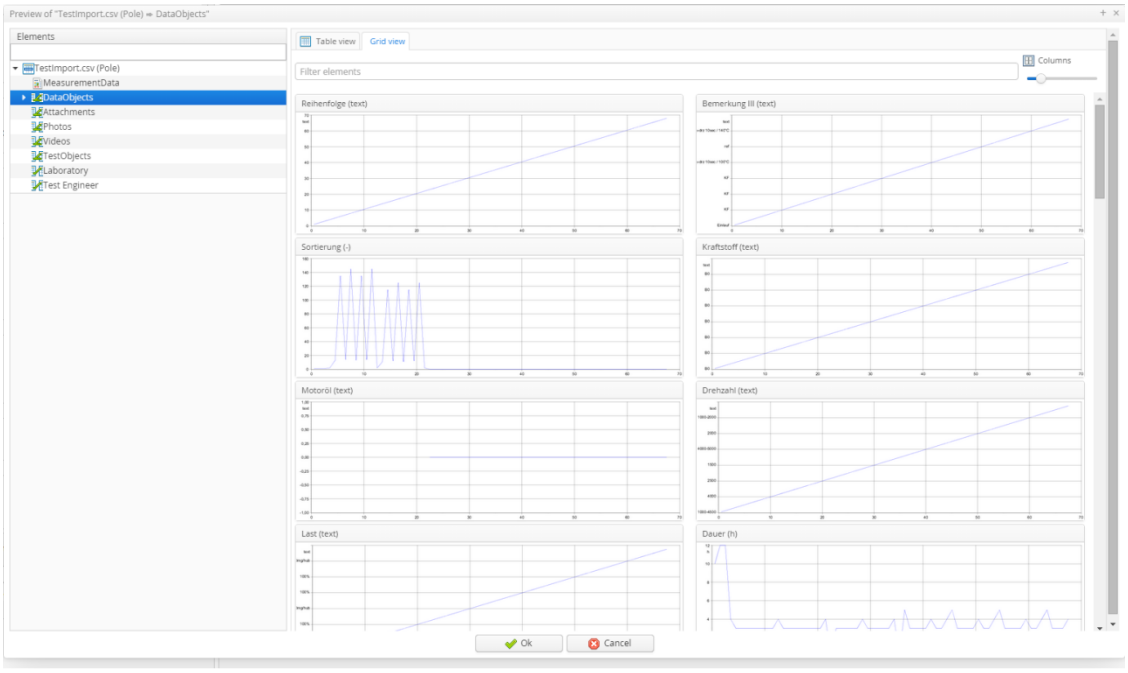


An overview of the data stored for a measurement can be viewed by opening the preview dialog. This can be done by either double clicking the measurement in the search table, or by clicking the graph icon in the search or selection tables.

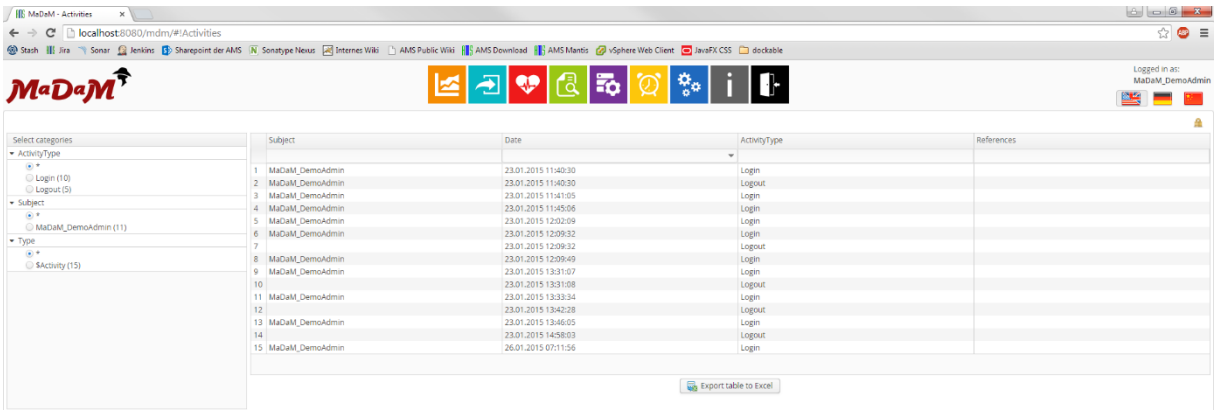
The preview dialog contains information about the access rights of a measurement, its metadata, the file structure that is stored within the system and all contained objects. For instance, DataObjects – i.e. channels – are included in this preview as well as Photos and Videos for Crash Tests. Exactly what is included depends on the type of the test and the configuration of *MaDaM*.

The tree in the left half of the dialog can be used to select what information a user wants to see. It is possible to preview a list of sub-elements, like channels.

Usually, a list of elements is displayed in a manner similar to the search table – as list of multiple columns displaying all available information. However, for certain elements like channels or photos, it is possible to open a 'GridView", displaying the images connected to the individual elements.

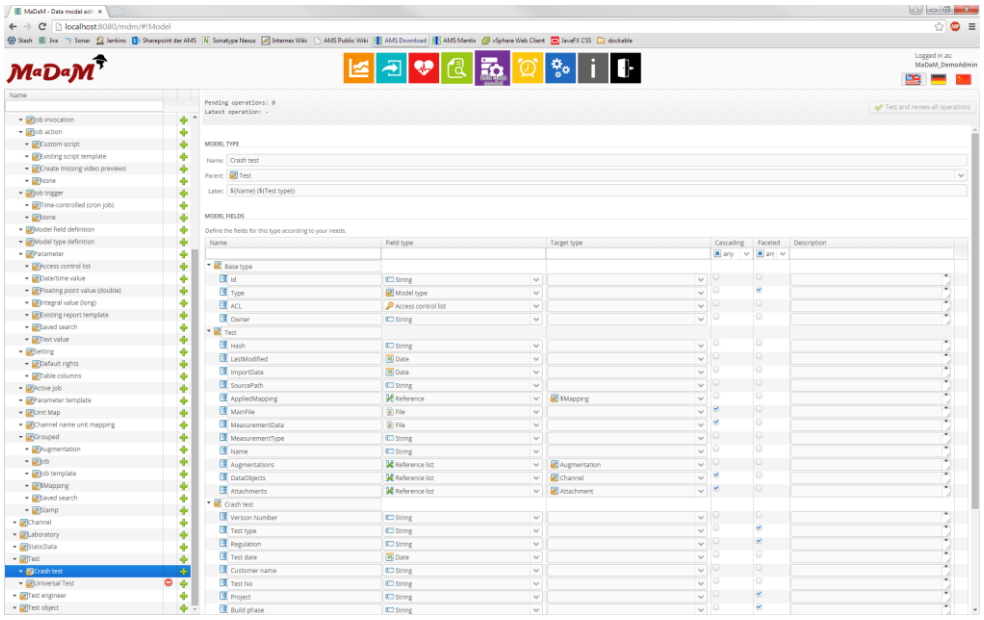


4.3.11 Activities

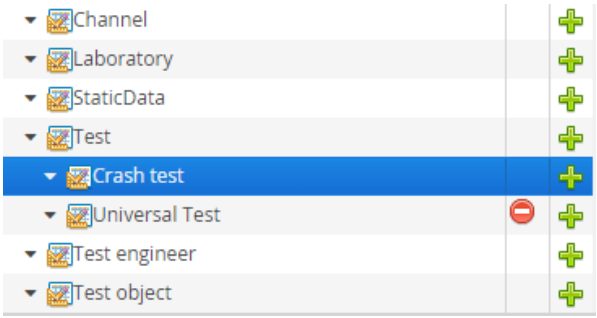


This particular page is a history of interactions with the system. It records logins, logouts, interactive analyses, pdf creations and data imports. Furthermore, the user who engaged in the action, the date of the interaction, and which references were affected are recorded as well.

4.3.12 Edit Data Model



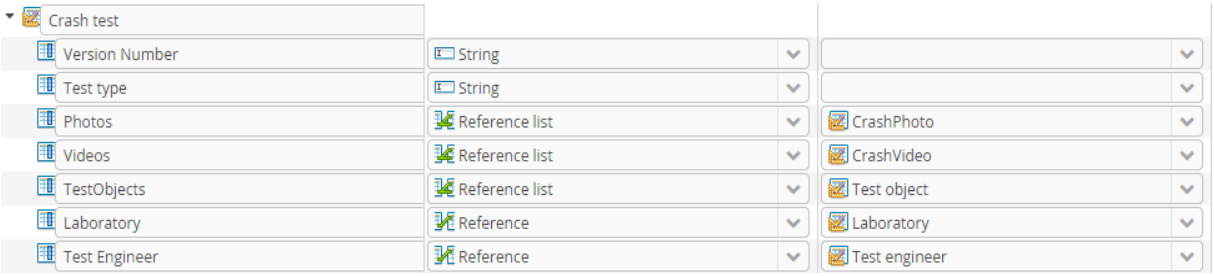
The ‘Edit data model’ view can be used to edit the configuration of the *MaDaM* system. On the left half, all the currently configured reference types (e.g. what kinds of test and what these tests contain) are defined.



In the example to the left, there are three tests defined: Test, Crash test and Universal Test.

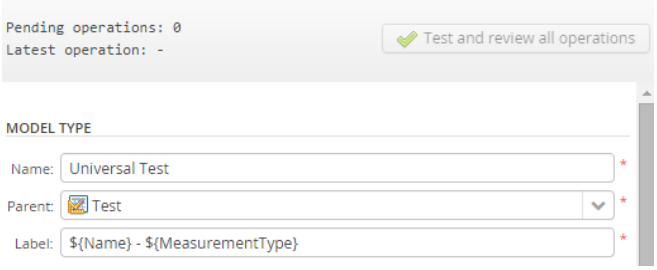
It is also apparent that the current user is only allowed to add new subtypes or remove the Universal Test type. This behaviour is due to the fact that the general base configuration of the system must not be changed in order to run properly.

Furthermore, there are other reference types defined: Channel, Laboratory, StaticData, Test engineer and Test Object. All of these specify a type of information that can potentially be stored in a test.

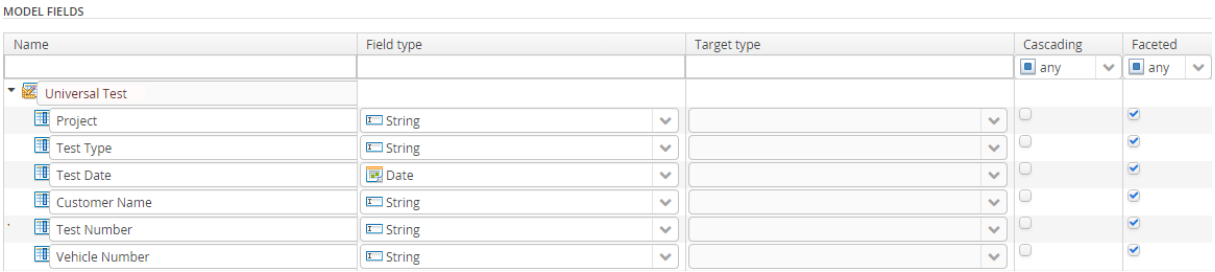


E.g. any Crash test contains a list of references in a field called 'TestObjects' (defined in the leftmost column), which are of the type 'Test Object' (defined in the rightmost column). It also contains a single reference in the field 'Laboratory' of the type 'Laboratory'.

It is important to note that these fields are inherited by subtypes. Therefore, any type that is a subtype of Crash test automatically contains all fields that are defined in Crash test and the parents thereof.



There are manifold ways to change the data model. First of all, it is possible to add subtypes to existing reference types. A user can change the name of a reference type and also change the parent type – which in turn changes what fields are inherited. Furthermore there is the option to change the label of a reference type. Changing this results in other information used to describe a reference of this type. In the above example, any universal test would be labelled according to its 'Name' and its 'MeasurementType' fields within MaDaM.



Every model field can have two modifiers: cascading and faceted. Cascading fields are always tied to the exact measurement that contains them. For instance, if a measurement is deleted and has cascading fields, these are deleted as well. All modifications on a measurement are propagated to its cascading fields.

The 'Faceted' checkbox decides whether a field is displayed in the facet tree or not. Only fields that are set to be faceted appear in the tree. Commonly, faceted fields should be the most discriminating fields for any one test type.

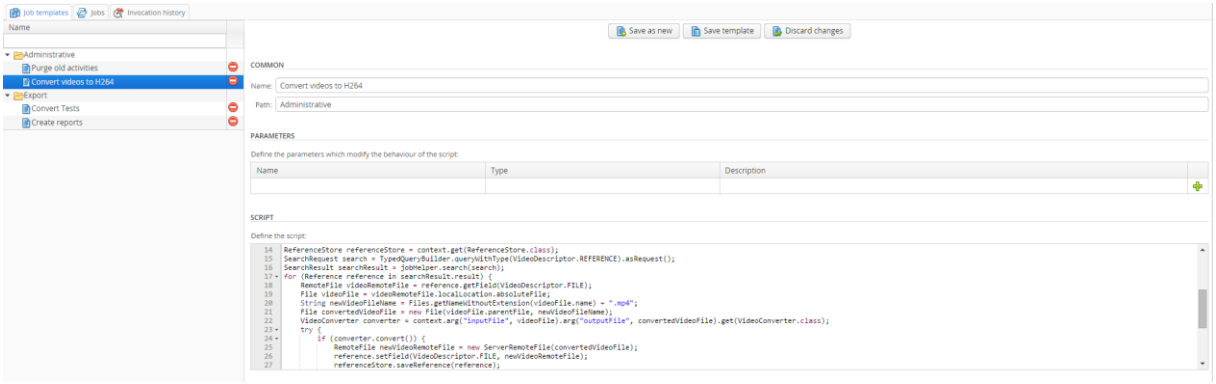
Fields can be added by clicking the '+' icon in the bottom of the tests description. In the same manner, fields can be deleted by clicking the '-' sign. It is possible to change the name of fields and also to change their type via the dropdown menus on the left hand side of the table.

It should be noted that users can only add fields if they have the permission set by the authentication system. The same is true for editing or deleting fields.

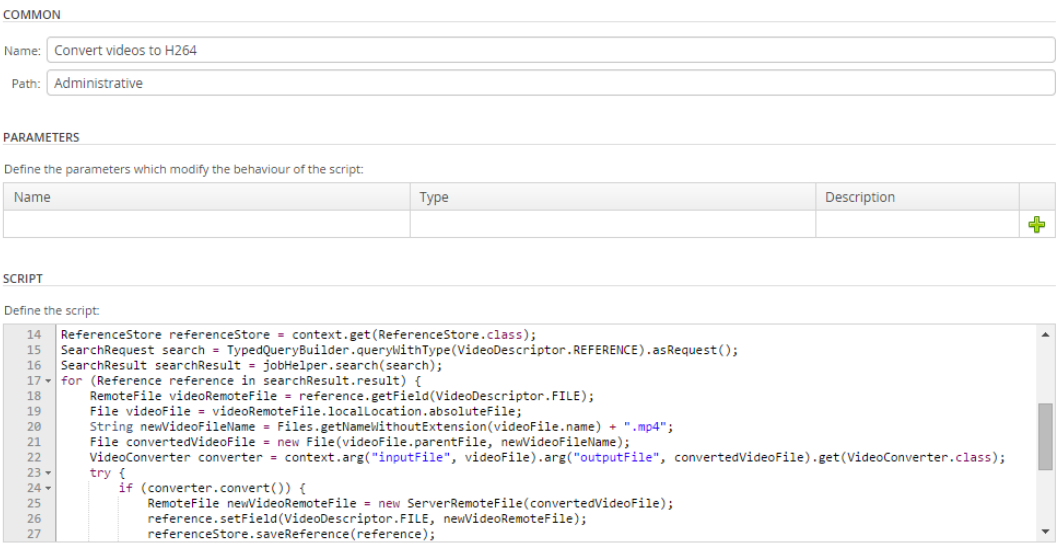
Any changes in the model need to be confirmed by clicking the 'Test and review all operations' button at the top of the 'Edit data model' page. Doing so opens a popup window giving the user information about the elements affected by this change. During this process, the changes are simulated to check whether they are valid or not. After checking for validity, the user can review the changes and is asked to confirm them once more.

4.3.13 Configure Jobs

Jobs are automated actions that can be configured to run at given intervals or be started manually. The three tabs are used to configure job templates, manage actual jobs or view the history of invocation of jobs.



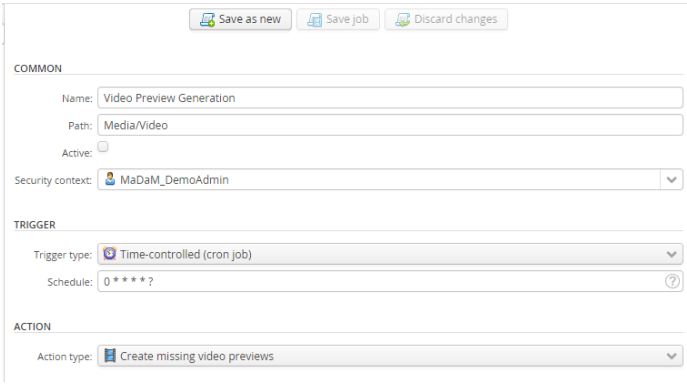
Job templates are scripts that can be used for the specification of jobs. While there are job templates for various actions predefined – e.g. converting all videos to H264 – no job is defined in the demo version.



Users can change the existing script and either save them as a new template or override the existing one that has been edited.

Such templates always consist of a name, a path, parameters and the script that uses these parameters. The name of a template is simply the name used in the MaDaM system and the path describes the type of job it should do in a hierarchical way.

The script defines the code that is executed when the job is invoked, and the user can define parameters that are available to the script.



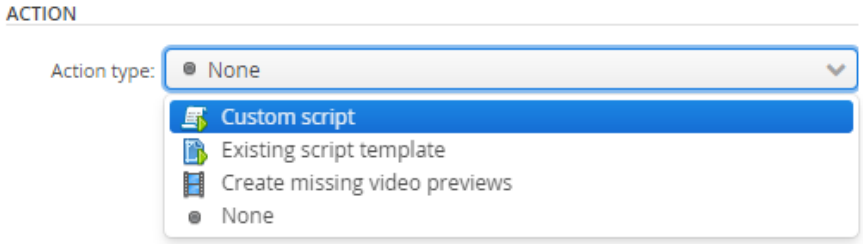
The second tab, labelled 'Jobs', shows the jobs that are currently defined. Editing the jobs is equivalent to editing the job templates regarding the use of 'Save as new' and 'Save job'.

When editing a job, users can change the name and the path of a job and define whether it is active or not. Every job is bound to a trigger type – either manual or time-controlled. However, the trigger only invokes the job if it's set to active.

The given example is time-controlled. Such jobs are referred to as 'cron job', and have a schedule to which they adhere. This particular example would trigger whenever the second of the system time of the server is '0' – meaning every full minute.

The syntax for time triggers is complex, and user guidance is offered by the question mark button in the 'Schedule' text field.

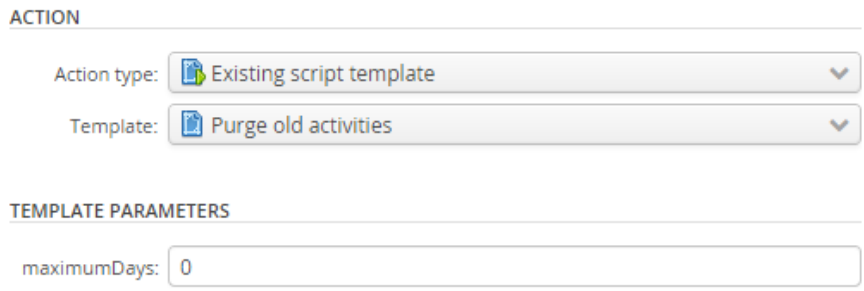
Furthermore, users need to specify a security context. The security context of a job defines which access rights it can use – for instance whether it can create new tests or change the existing data.



Finally, there is the action type. This defines what kind of action is executed when the job is invoked.

The 'Custom script' option shows a text field into which a groovy script can be entered. This script is then executed and is not stored as a template.

Choosing 'Existing script template' offers the user to select any of the existing job templates and specify script parameters.



The example to the left uses the 'Purge old activities' template which maintains the 'Activities' log of MaDaM.

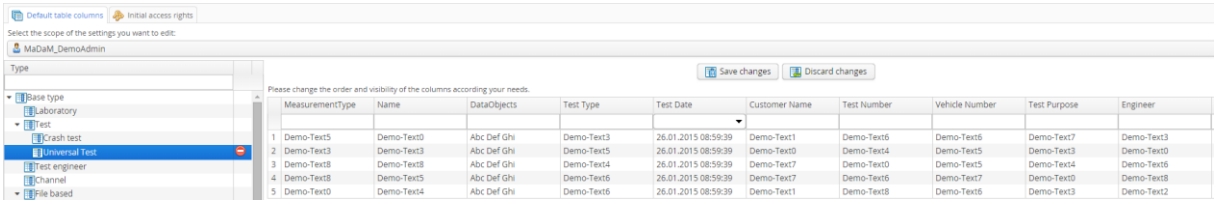
The user can define how long activities are stored. If they are older than the 'maximumDays' parameter allows, they will be deleted by the job.

Finally, there is a special 'Create missing video previews' action. This kind of job is intended to be used in environments where videos are commonly connected to tests – for example crash tests. The job creates the preview images that are used to discern which video is included in a test.

Jobs with the 'Create missing video previews' option extract these thumbnails and store them in the system. In doing so, it checks all videos stored in MaDaM and checks whether they already have a preview image.

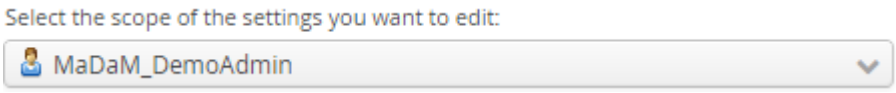
4.3.14 Settings

The 'Settings' page is used to define two different aspects of the MaDaM system.

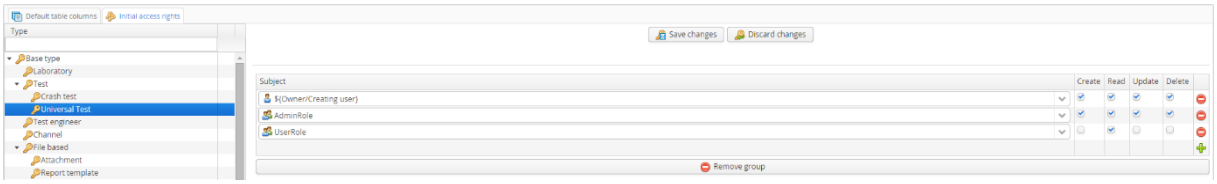


Firstly, the default table columns can be defined. These are the columns that are shown in the 'Create analysis' and the 'Reference view' pages when this type of reference is shown.

Configuring the columns includes changing which columns are shown at all by going to the rightmost corner of the table and clicking the small wheel button. Furthermore the order columns are shown in can be customized.



The changes regarding the column configuration are always bound to a scope. The scope defines what user group is affected by the settings.



Secondly, the default access rights can be defined. It is of note that these configurations are passed on to subtypes. The access rights definition in 'test' also applies to 'Crash test' and 'Universal Test'. If references of a certain type do not specify their own access rights, the system default will be used instead.

The access rights follow the CRUD paradigm. Therefore the system differentiates four basic interactions: create, read, update, and delete.

In the above example, both the creating user and users from the 'AdminRole' user group are allowed to create, read, update and delete tests of the type 'Universal Test'. Users belonging to the 'UserRole' rights group are only allowed to read such tests.

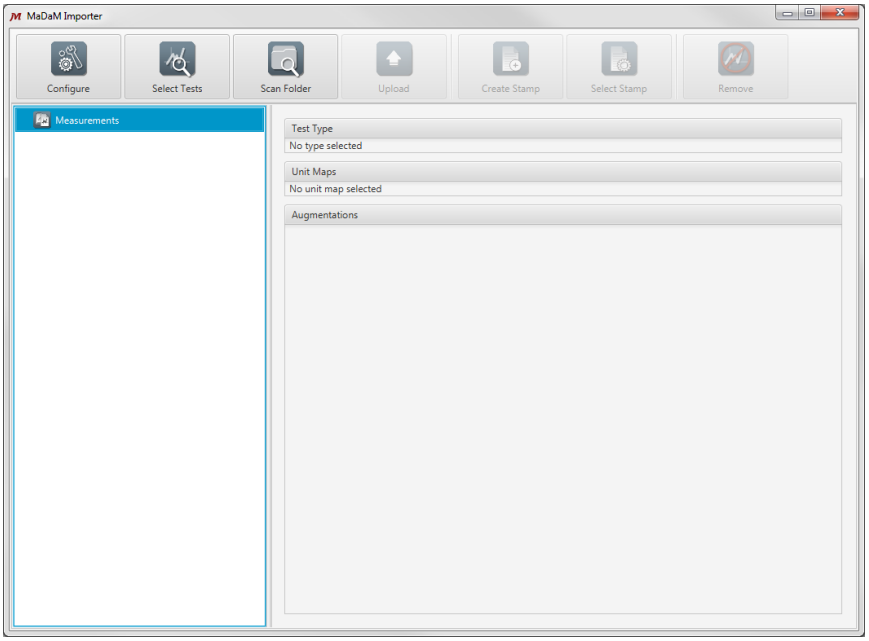
The roles available to the system have to be preconfigured via the 'MDM-Configuration.xml'. It is purposely not explained how to do this, since this is irrelevant for demo purposes and is done by the AMS a priori.

4.4 Measurement Importer

To import measurements, a user needs to click the 'Import Measurements" button in the navigational dock. Doing so triggers a JNLP download, which in turn starts the MaDaM Importer.

It should be noted that the look of the components depend on what Java version is installed on the client. The images provided in this documentation were created in a Java 8 environment. However, while the look may differ, the functionality of the Importer and the way to operate it stay the same.

4.4.1 Layout



The Importer itself consists of three major parts:

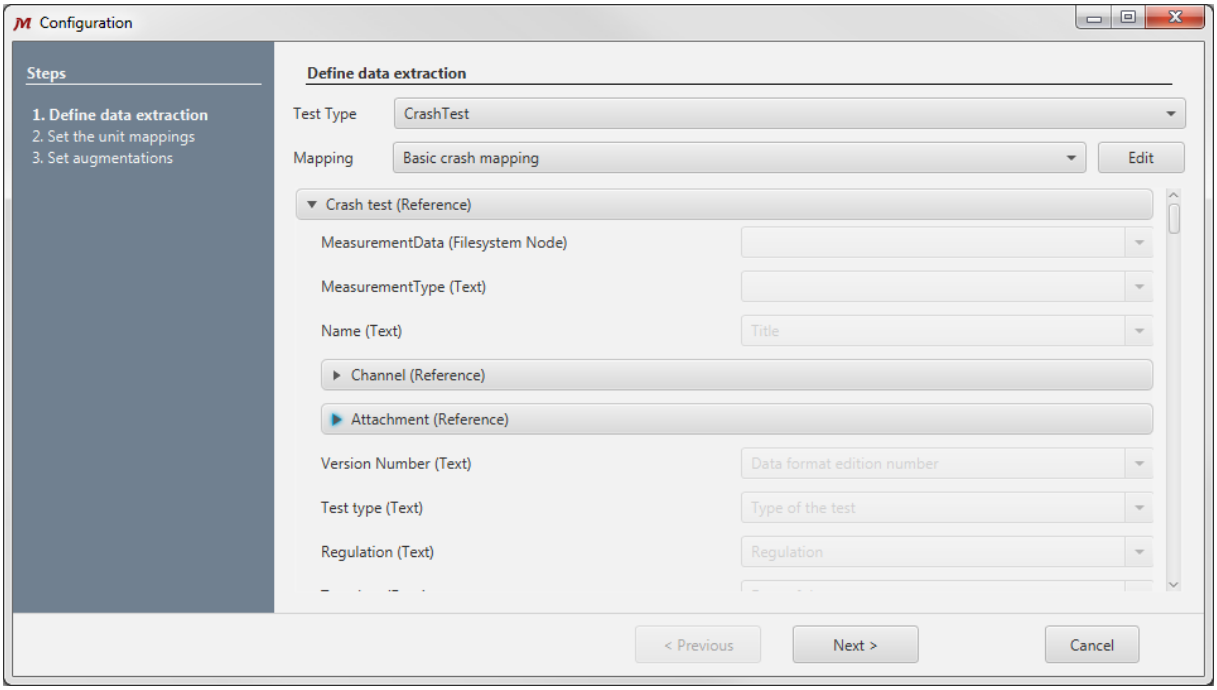
- **Toolbar** - the toolbar at the top of the importer
- **Measurements tree** - the left half of the importer (which is empty in the above image)
- **Content view** - the right half of the importer, displaying information about the selected item

4.4.2 Configuration

To guide the user through the process of importing measurements, there are certain aspects that need to be defined in the system. It is necessary for the Importer to know what kind of test is imported and how the information should be extracted from the original measurement files. Furthermore, there are optional setup steps that will be shown in detail.

Defining the necessary information can be done in a setup wizard, which is opened by clicking the 'Configure" button in the toolbar.

Define Data Extraction



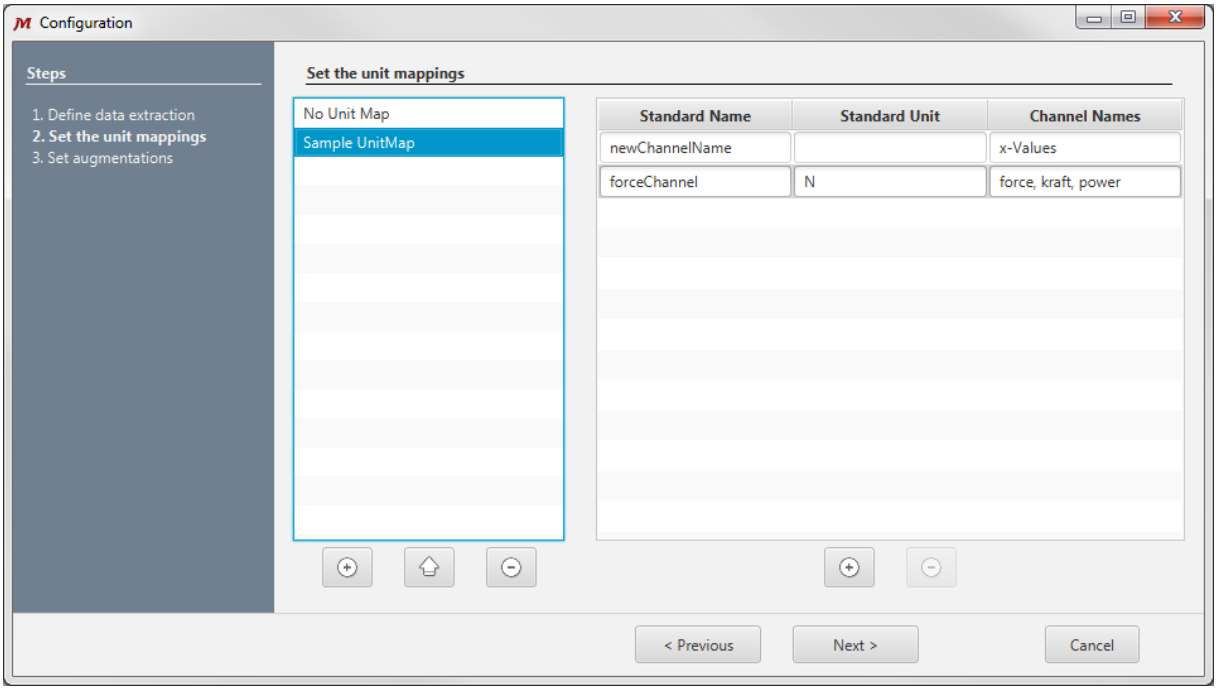
The first setup step is ‘Define data extraction”, a step where the user defines what kind of test he is importing and what data should be extracted. For demo purposes, there is one kind of test preconfigured - an MME based crash test.

There are two dropdown menus in this step: the ‘Test Type” box, in which the user needs to select the type of the test. The selection in this box is dependent on what kind of tests are configured in the MaDaM system and what kind of tests the user is permitted to created. The configuration itself defines what kind of fields a test should have – it is the format which will be used by MaDaM.

The second dropdown menu allows the user to select a ‘Mapping”. This mapping specifies what information from the original measurement should be assigned to the MaDaM specific format. The above example specifies that the measurement contains a field ‘Title”, which should be assigned to the ‘Name” field of MaDaM Measurements. Furthermore, the original file contains a ‘Data format edition number”, which should be assigned to the ‘Version Number” field of the resulting Crash Test.

The specification of custom mappings requires more extensive training. However, demo deployments of MaDaM come with exemplary settings which can be used for evaluation.

Set the Unit Mappings



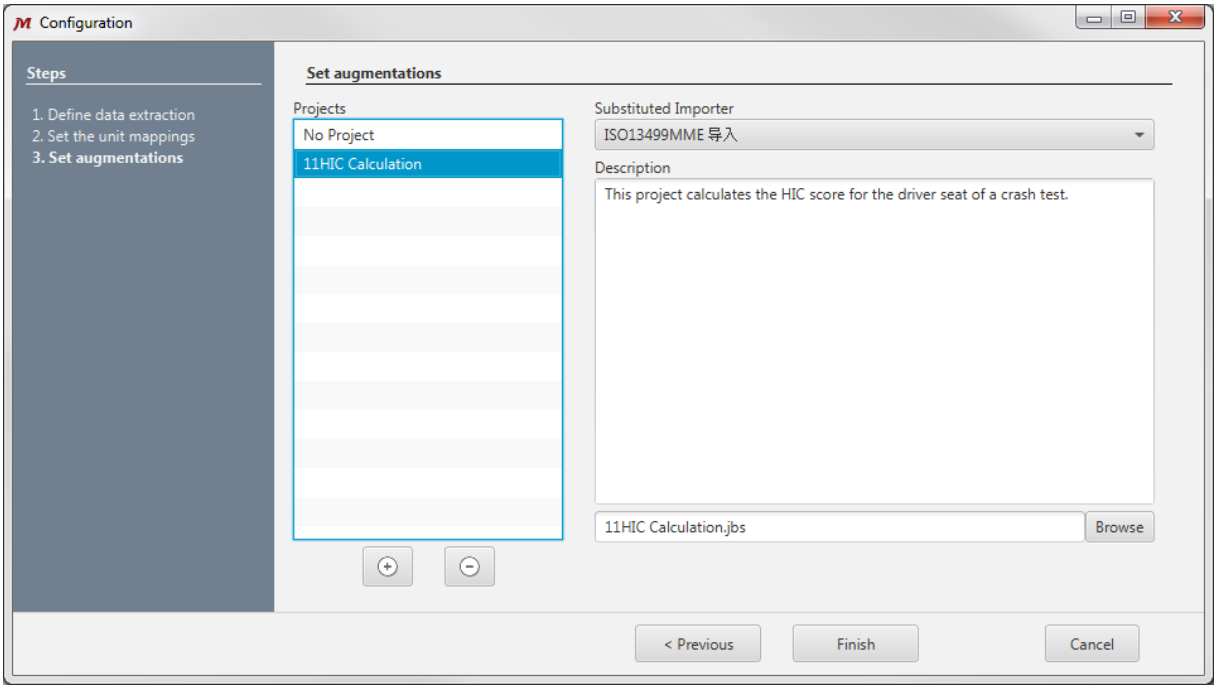
The second setup step is used to set channel-name-unit-maps. In the above example, every channel named 'x-Values" would be renamed to 'newChannelName"; and all 'force", 'kraft" and 'power" channels would be renamed to 'forceChannel" and have their unit recalculated to be Newton. It is still possible to import measurements where none of the channels match the map. Furthermore, it is entirely possible to skip this step by simply selecting 'No unit map".

Users can either create new unit maps, import existing *.cmap* files or remove unit maps by using the three buttons below the first list in the above example. The second half of the dialog shows a table that lists all of the entries of the selected unit map. New entries can be added by clicking the '+" button and old entries can be removed by clicking the '-' button. The name of a unit map can be changed by double clicking it in the list and pressing the return key to save the changed name.

Set Augmentations

The final step of the setup is used to set so called 'augmentations". These are jBEAM projects that are used calculate additional values in the form of virtual channels. The control itself is similar to the unit mapping step of the setup, in that it has a list of the available projects and a part for more details on the selected project.

Just like unit maps, it is possible to add or remove projects by clicking either the '+" or the '-' button and projects can be renamed by double clicking their entry in the list. When trying to add a project, the user is prompted to select a *.jbt* or *.jbs* file, which contains the actual project.



When adding a new project, the setup wizard checks whether there are still importers in the project file that are used to calculate channels. If there is such an importer, the user is forced to select a 'Substituted Importer", meaning that he decides what the importer should be replaced by the measurement imported in the MaDaM importer.

In the above example, the project file contains an 'ISO13499MME" importer for its calculation of the HIC score. However, when a user imports a measurement with the MaDaM Importer, the HIC score should be calculated based on that measurement. Thus, the 'ISO13499MME" importer that is still in the project is replaced by the measurement imported with the MaDaM importer.

Similarly to the second step of the setup, this is completely optional. The user can simply select 'No Project" and finish the setup.

4.4.3 Importing Measurements

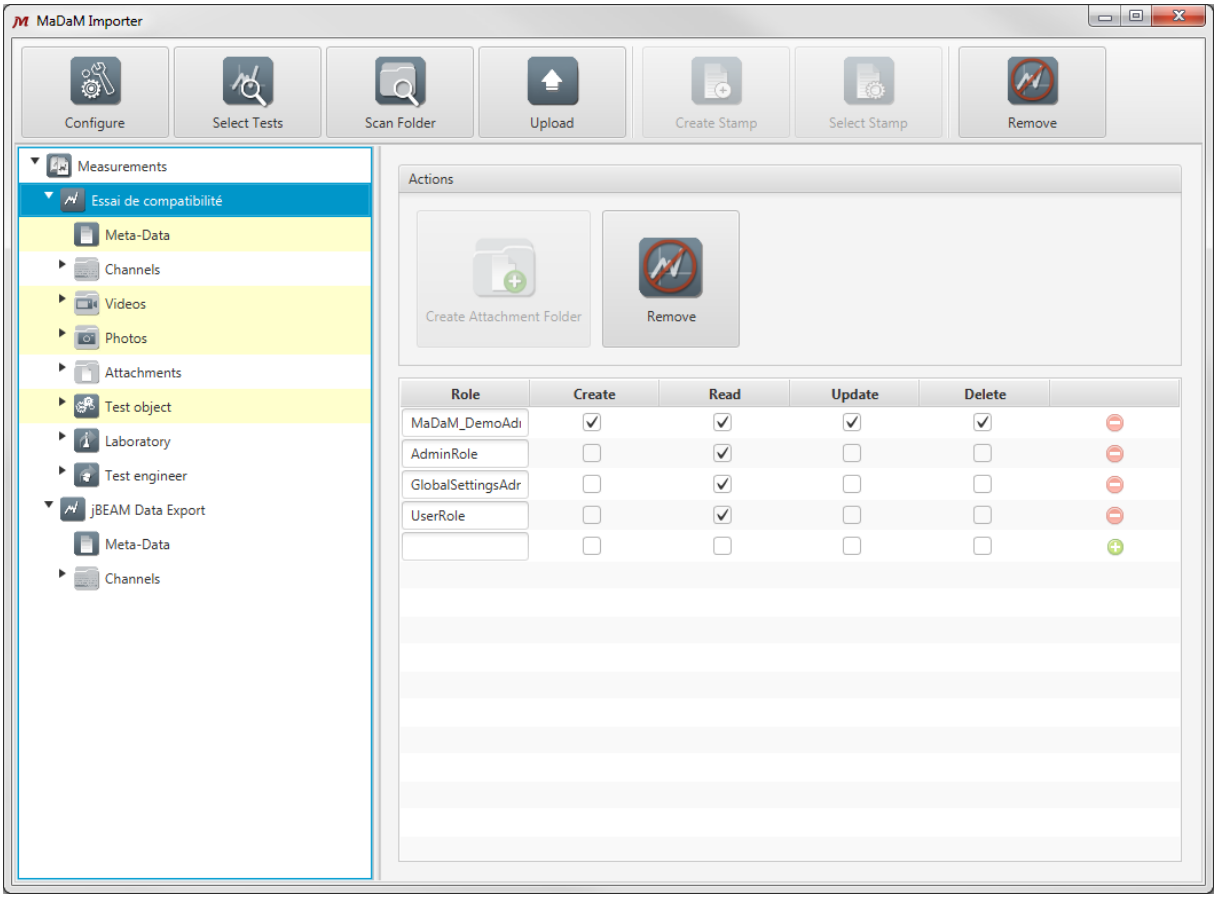
After finishing the setup, the user is free to import measurements. There are multiple ways to do this.

The toolbar offers two ways to import data: the 'Select Tests" and the 'Scan Folder" button. Clicking 'Select Tests" opens a file selection window that allows navigating to a folder and selecting files from this folder. 'Scan Folder" works differently. A similar dialog is opened, where the user can navigate to a directory. Instead of selecting individual files, however, a user selects a complete directory. The MaDaM Importer tries to find measurement files in the select folder and every single subfolder contained within it. This allows for importing larger quantities of measurements in one go.

The last and most comfortable option is to simply drag and drop a measurement file into the left half of the Importer.

Should a user try to import a measurement without configuring the system first, a notification will force the user to either start the setup or discard the data that should be imported.

The process of loading is made visible by a progress bar that is shown in the Importer. After loading, the measurement is shown in a tree structure. Again, the elements shown depend on the selected type and the fields this type has been configured to have in MaDaM.

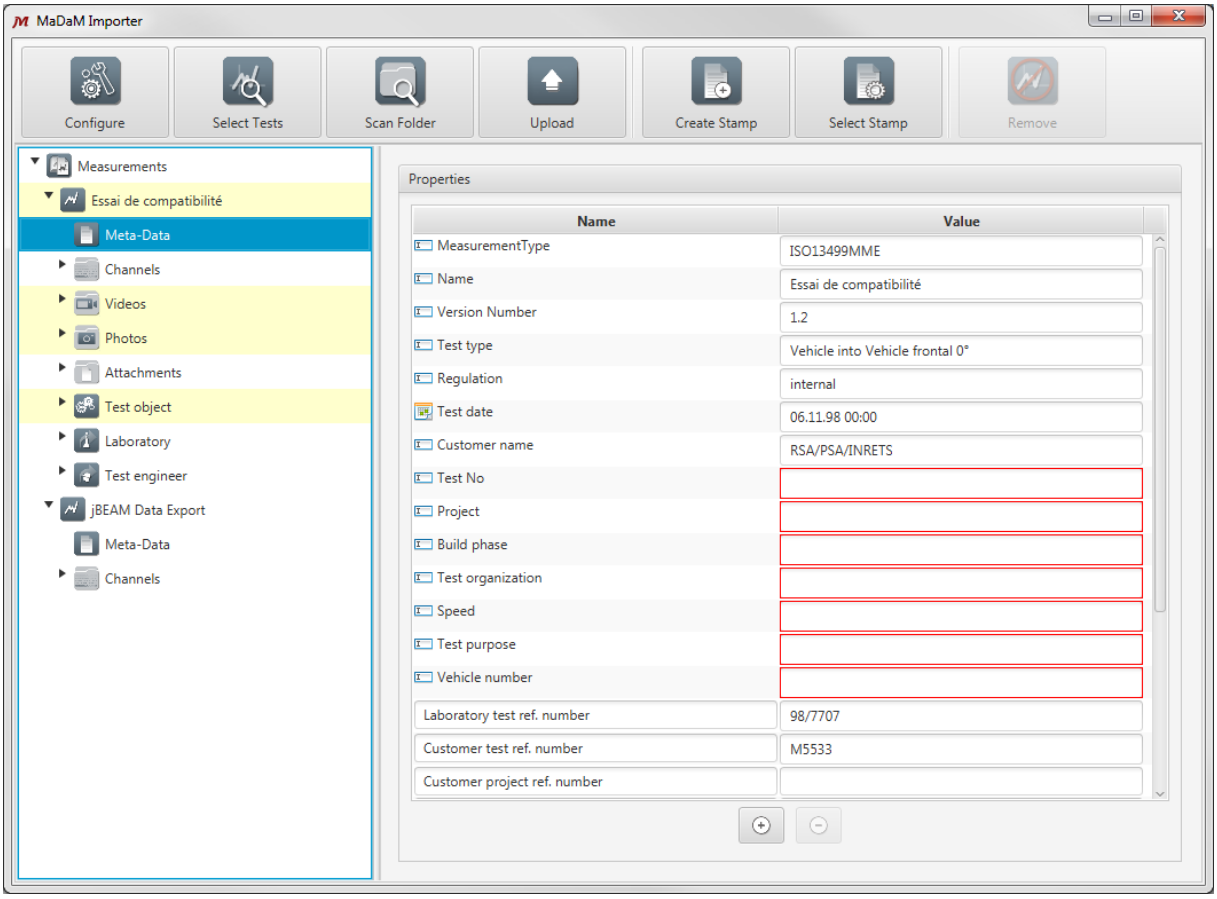


The above example shows one *CrashTest* at the top, labelled 'Essai de compatibilité" and a default *Test* labelled 'jBEAM Data Export' at the bottom. The differences are clearly visible, as the only elements every imported measurement must have are 'Meta-Data' and 'Channels'. The other elements, like 'Videos' and 'Photos' are configured to be part of *CrashTest.*

To store the imported measurements in ***MaDaM***, a user simply needs to click the 'Upload' button. However, this is only possible if the measurements meet certain requirements.

4.4.4 Validation and Uploading

As it is the very purpose of the importer to enforce a uniform makeup of the data stored in ***MaDaM***, it checks every measurement for validity. The demo deployment contains a basic validation that checks if the information needed by the system exists, is complete and in the correct format.



The above example shows information about the test itself, the Meta-Data. There are two types of information contributing to this particular set of Meta information: *model information* and *additional information*.

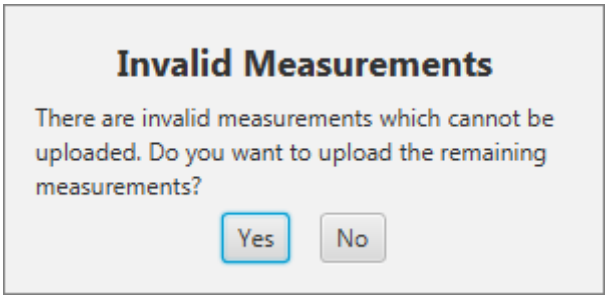
The *model information* is the one that is required as per the definition of the imported test type. It is depicted by the labels that have an icon depicting the type of information needed – e.g. text or a date.

If there is no information connected to these fields, they are coloured red. This also applies to information in the wrong format. An example for that would be entering a text while expecting a date or a number.

Should a model field be empty or contain information in an incorrect format, the element that holds this model field is marked as invalid. At the same time, the measurement to which this element belongs is marked as invalid, too

Additional information is the type of information where the field name on the left side can also be edited. They are optional, and can be used to convey additional, but nonessential information about the measurement.

If any of the measurements that are currently imported do not meet the validation criteria, a prompt inquires whether the user wants to upload the valid measurements or completely abort the uploading process. This also serves as an information about invalid states.


Invalid Measurements
There are invalid measurements which cannot be uploaded. Do you want to upload the remaining measurements?
Yes
No

4.4.5 Editing Measurements

Measurements can be edited after being imported to achieve a valid state. It is important to note that none of the original files are modified in any way, with MME Crash Headers being the only exception. This means that if a user changes the name of an image in the Importer, he only changes the name that is stored in MaDaM and not the actual filename.

As has been stated before, what kind of elements are created for a measurement depends on the configuration of the MaDaM system and the type of the imported measurements. However, there are six general types of elements that can be contained within a test in various ways:

- Meta-Data
- Channels
- Photos
- Videos
- Attachments
- References

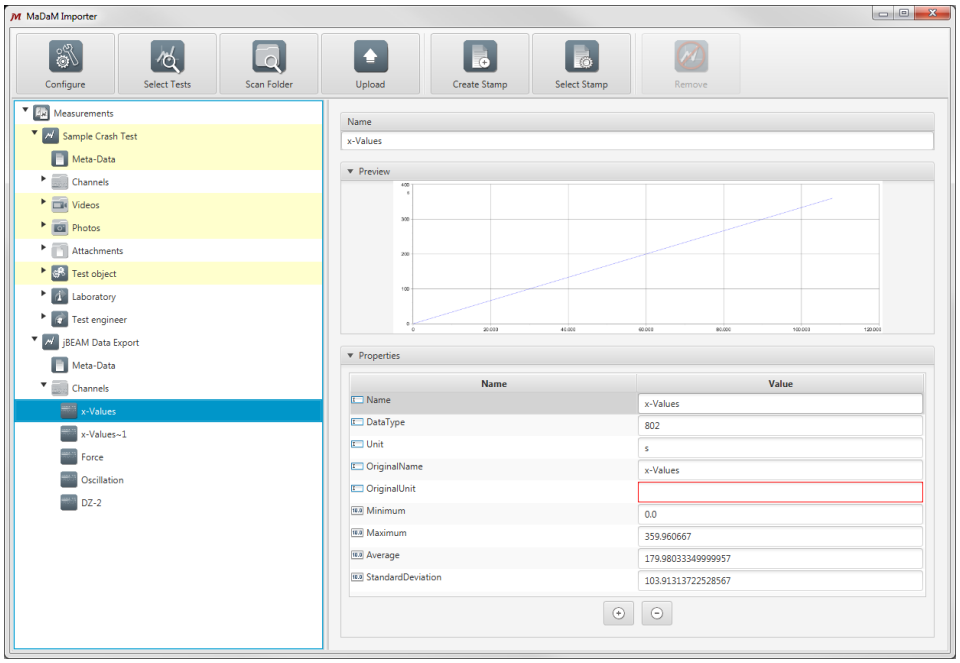
When selecting an element in the left half of the importer, the right half is filled by a *control*. A control is a graphic element displaying information and allowing the user to interact with the presented data.

The only similarity, that almost all controls share are properties. A property consists of a name and a value and it represents information about the element it is connected to. The controls for properties look very much alike. It is a list, containing a '+' button to add new properties and an '-' button to remove existing ones.

Meta-Data

The Meta-Data of a measurement reflects the general information about the imported test. This includes a set of model fields and additional fields as well. Selecting the Meta-Data node of a measurement displays this information in the form of a table. For Meta-Data, it is always possible to edit the value of any field. It is furthermore possible to change the name of non-model fields – i.e. fields that are non-essential to the measurement.

Channels

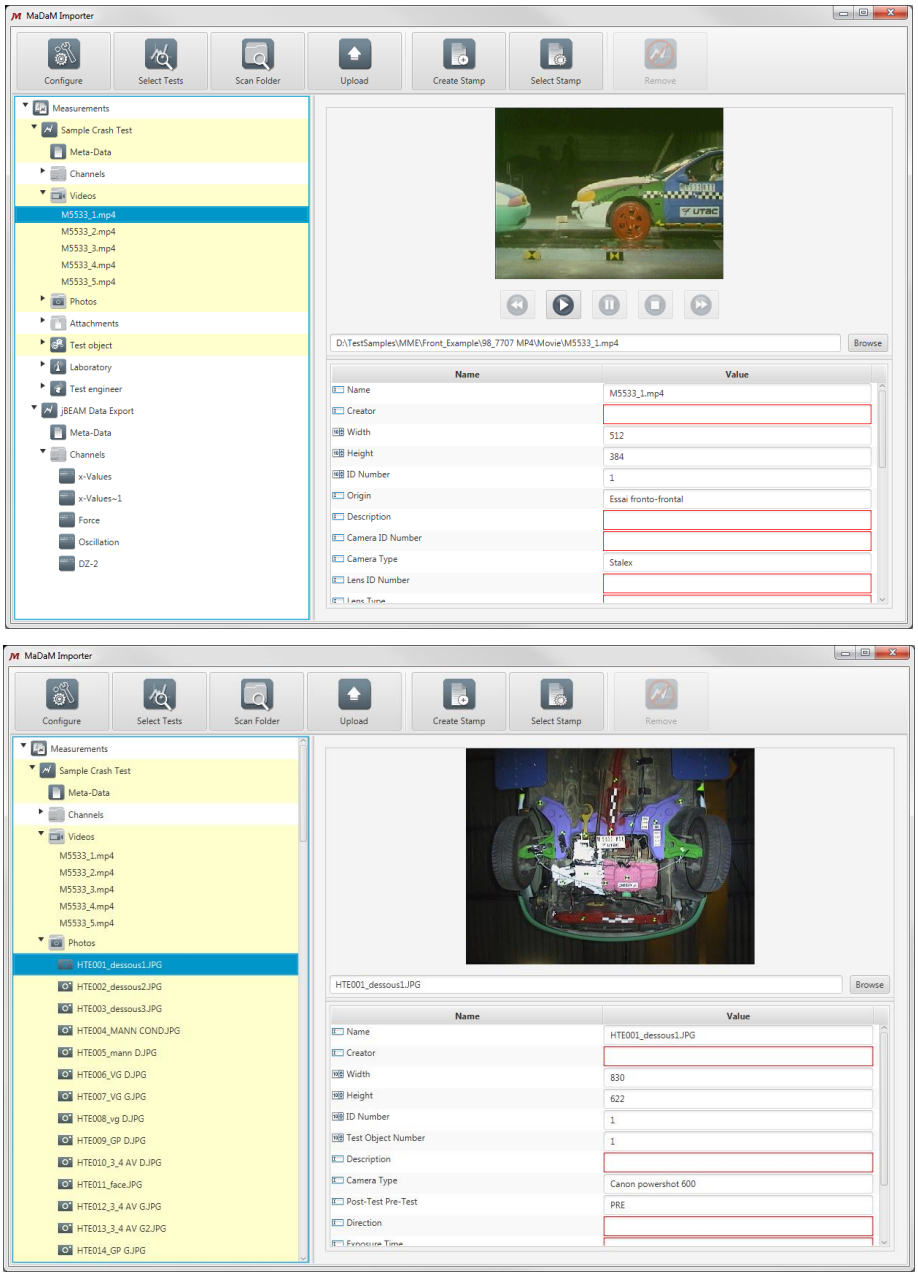


Selecting a channel element displays various kinds of information. On the top, there is a text field that can be used to edit the name of a channel. The middle shows a preview of the channel, and the bottom displays the properties of the channel. While this looks the same as the property display of other elements, the user cannot change the values of the model fields for channels.

While it is possible to add new properties and edit them freely, a user cannot simply change the average or the maximum of a channel. This is simply due to the fact that channels consist of statistical, and therefore unchangeable, information.

The name of a channel poses the only exception to this rule. While it can be changed via the text field at the top of the control, the original name is always stored and cannot be changed.

Videos and Photos

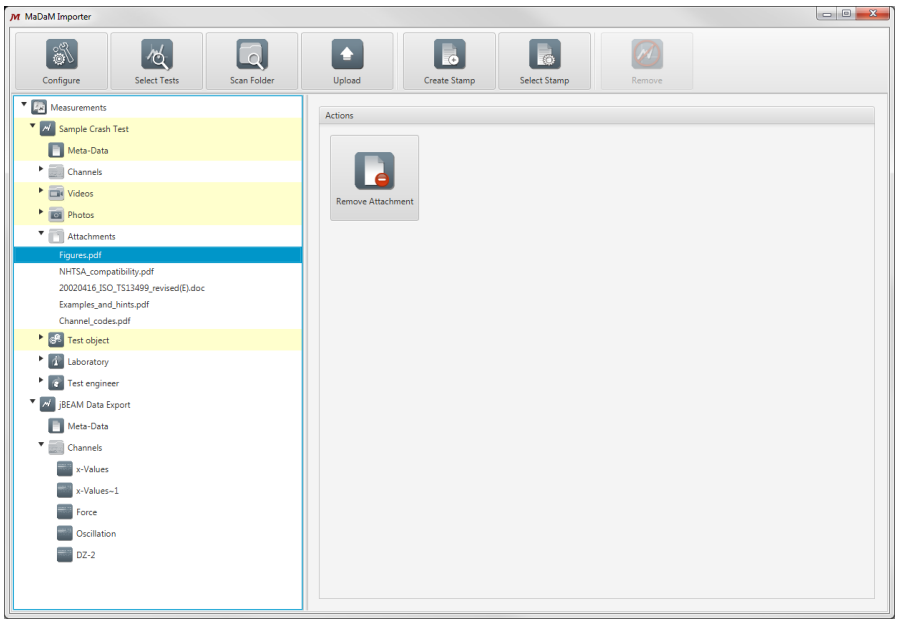


The controls for photos and videos are very much alike. They are separated horizontally, with the top displaying either the photo or the video, and the bottom displaying the properties.

The top half of the control shows the content and also the path to the currently selected file. By clicking the ‘Browse’ button, the user can change the file. This automatically reloads the displayed image or video and changes the name of the element in the measurement as well.

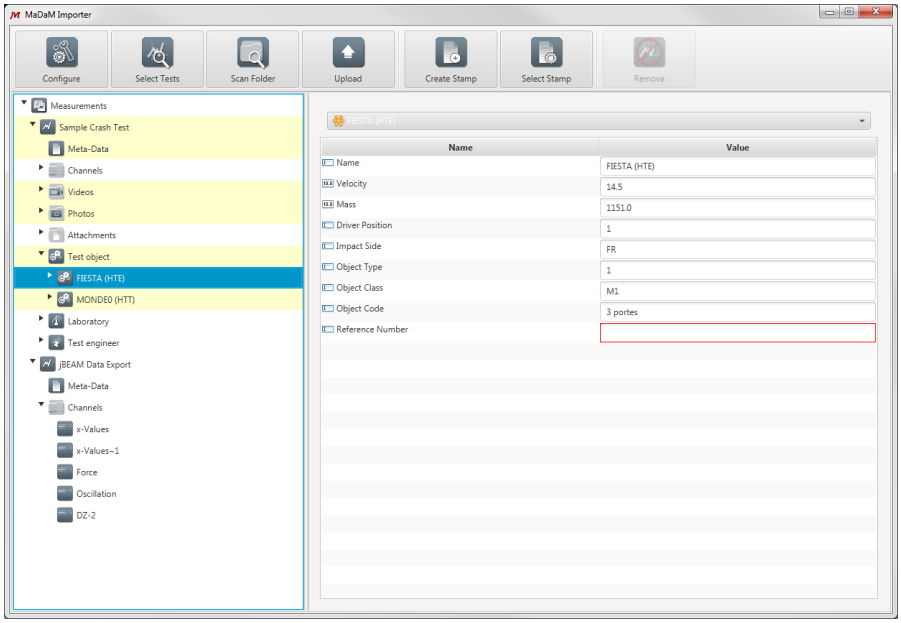
It should be noted that the videos can also be played in the *MaDaM* Importer, to ensure that the correct file is selected.

Attachments



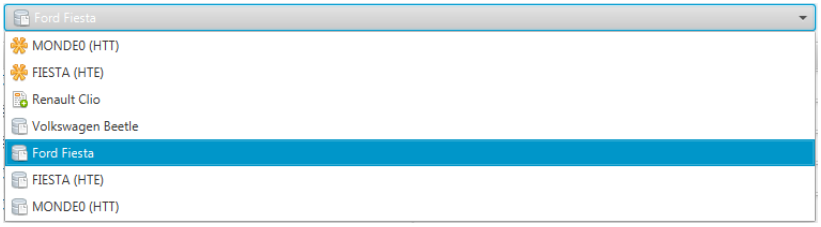
Attachments are files of any format that can be attached to a measurement if so configured. There are no properties connected to attachments, which means that there is nothing that can be changed by the user. However, it is possible to remove attachments.

References



References are complex objects that are contained in another element. For example, every *Crash Test* measurement includes a laboratory, a test engineer and any number of test objects. What differentiates them from other fields – i.e. properties – is that they have a definition of their own. There are certain fields that every test object needs to have, like a name, velocity and mass.

Every reference can contain properties and other references. The control that is created when selecting a reference from the measurement consists of two parts: a selection dropdown menu and the property view.



The dropdown menu can be used to select already existing references to ensure consistency. Consider the following scenario:

A test engineer wants to import the latest crash test in a series with the same model. Instead of entering the same information multiple times, he simply opens the dropdown menu and selects the entry for his car.

There are three different states in which a reference can be in this dropdown menu:

- New Reference - always creates a new reference, it has an orange star as icon
- Local Reference - selects a reference that was imported during this session
- Remote Reference - selects a reference that was stored in another importing session

Upon importing the data, the MaDaM Importer tries to match the references of each measurement with already existing ones in the data storage or other measurements imported in this session. This step is taken to ensure comfortable usage while still guaranteeing consistency with the data.

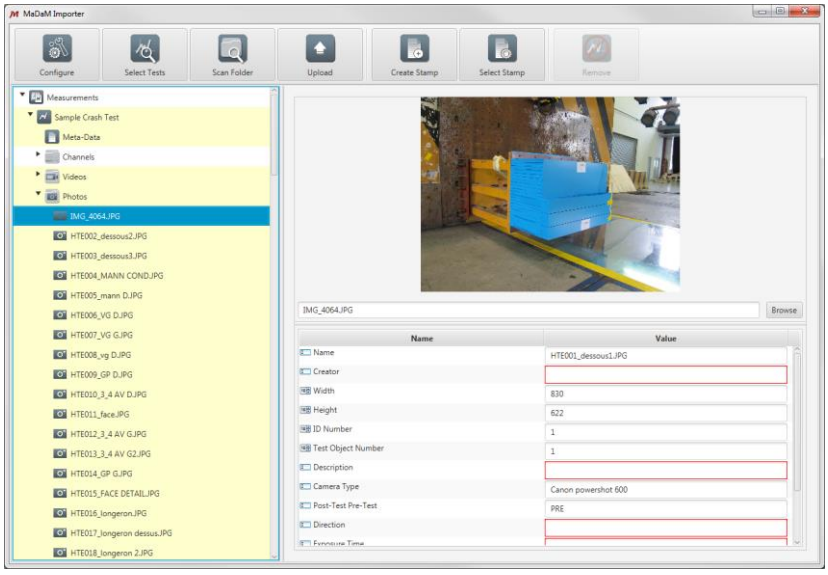
4.4.6 Stamps

Establishing validity for multiple measurements, or for measurements with a lot of contained items, can be time-consuming. Especially since often times, the missing information will be very similar.

To circumvent drawn-out editing processes, the MaDaM Importer offers a stamp system.

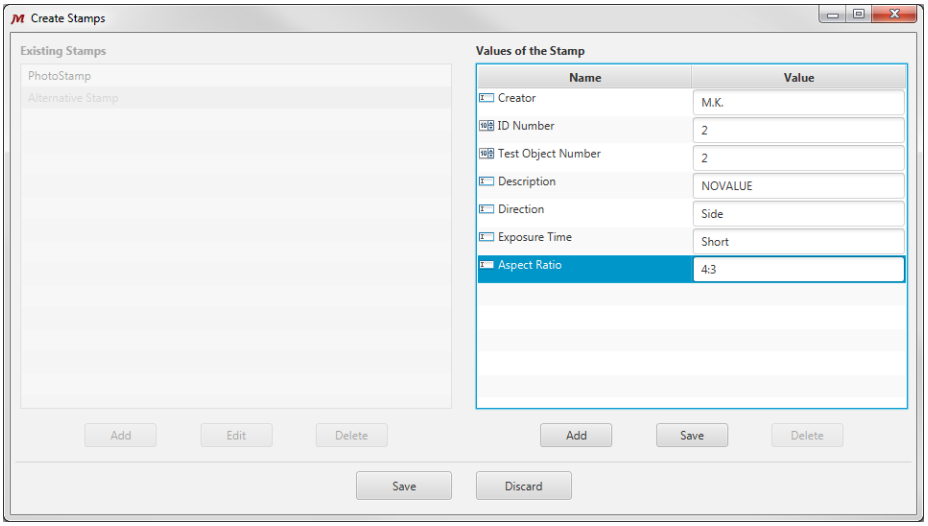
A Stamp is a collection of fields with a set value that can be applied to multiple items of the same type instantaneously. The user can create stamps via the 'Create Stamp' button, which becomes enabled when exactly one element is select to which stamps can be applied. Created stamps can then be applied via the 'Select Stamp' button, which becomes available as soon as one or more elements of the same type are selected – if this type allows for stamps.

For instance, the photos used to showcase the photo control under 4.3.5 all lacked certain fields – Creator, Description and others.



It is now possible to click the 'Create Stamp' to start creating stamps. In doing so, the user opens a window with two parts. The left half has a list of stamps, where new stamps can be created and existing ones can be

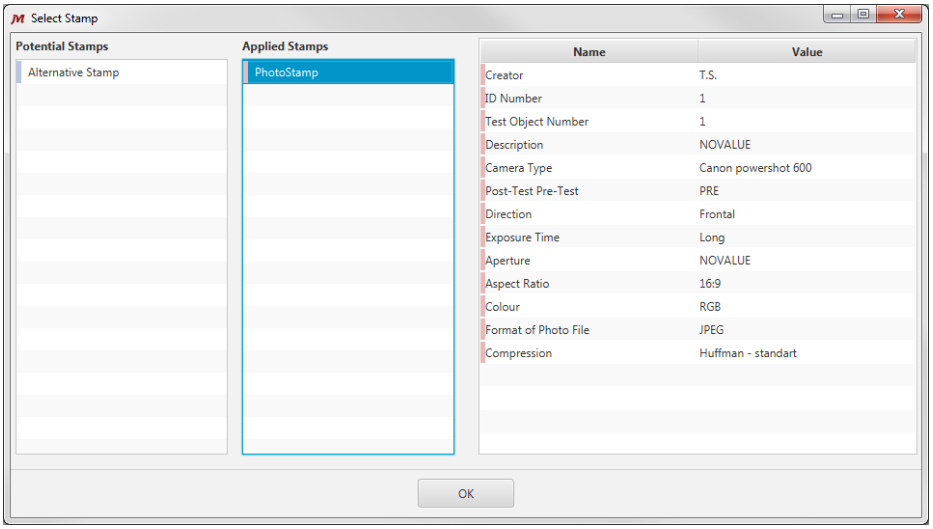
deleted. Furthermore, it is possible to rename stamps by double clicking them and then pressing the return key to confirm the new name.



In order to edit the properties connected to a stamp, the ‘Edit’ button below the stamp list needs to be clicked. This enables the right half of the dialog, where properties can be added or removed and their values can be changed.

The ‘Save’ button below the right half of the dialog saves the current changes, but does not close the window.

To save the overall changes regarding the stamps, the user needs to click the ‘Save’ button at the top of the dialog. If the ‘Discard’ button is selected instead, all changes since opening the dialog are discarded.



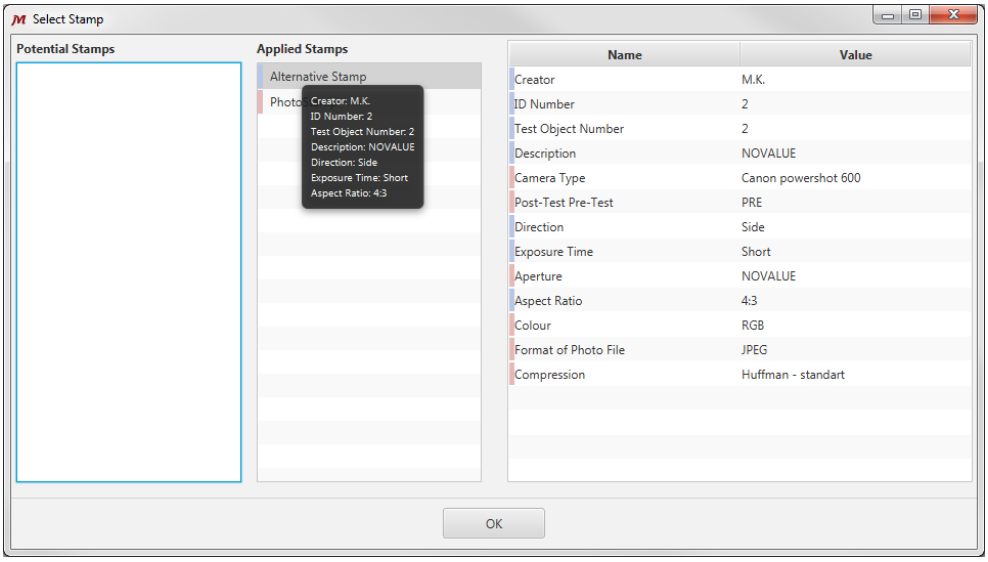
After finishing the creation dialog, the user can select all elements of that type unto which he wants to apply stamps. By clicking ‘Apply Stamps’ at this time, a new dialog is opened.

There are multiple unique features to the ‘Select Stamp’ dialog. First, the user can drag and drop stamps from the ‘Potential Stamps’ list to the ‘Applied Stamps’ list – meaning that he wishes to apply these stamps to all elements currently selected in the main view of the Importer.

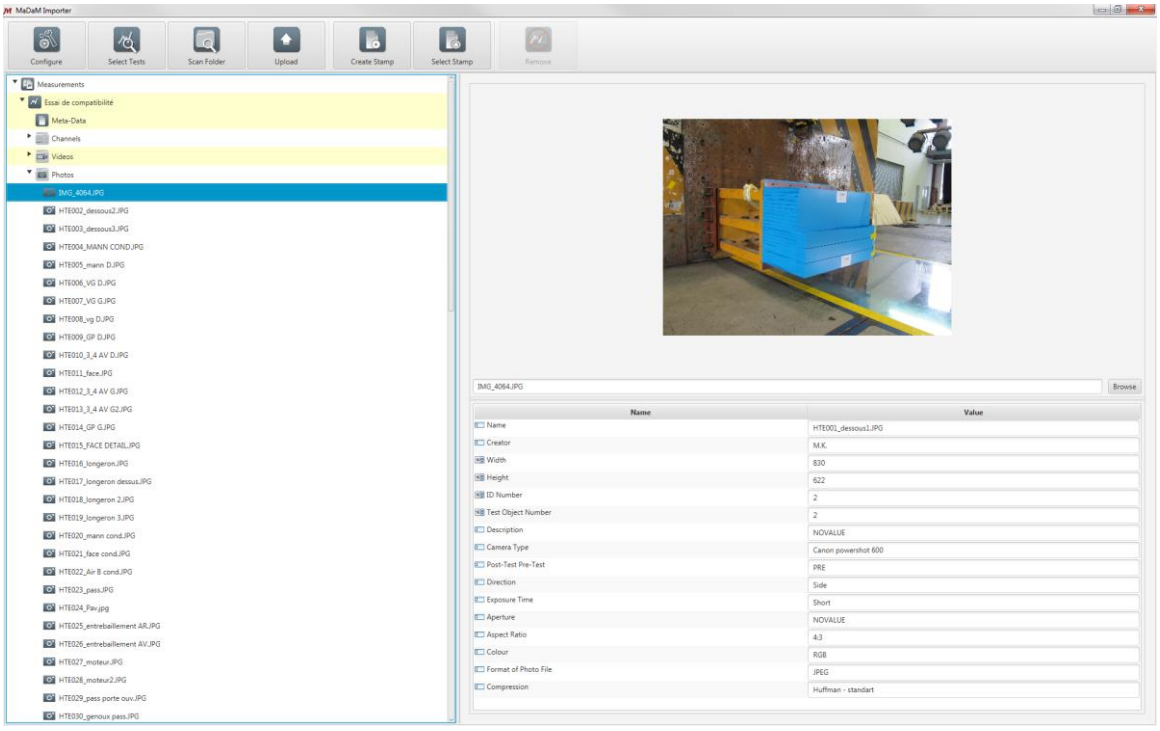
The right half of the dialog shows the resulting fields after combining all elements from the ‘Applied Stamps’ list. These properties are colour coded to show what stamp causes which value.

Furthermore, the items in the ‘Applied Stamps’ list can be sorted by using drag and drop to move them up or down the list. The stamp at the top is the most important one, meaning that the properties connected to it overwrite the other stamps.

An example of this can be seen below. While the red *PhotoStamp* already had the Creator field filled with the value 'T.S.', the blue *Alternative Stamp* overwrites this field with its own value. All fields that are contained in both selected stamps are overridden in this way.



Selecting the 'OK' button in this dialog applies the 'Applied Stamps' list to all items that are currently selected in the main view of the importer. This is the same as selecting every element individually and entering the information from the stamp manually. The result of this operation is shown below.



Stamps are stored in the main MaDaM system and can therefore be used in any Importer session after they have been created once.

While faster and more comfortable editing is the main focus of stamps, they can also be used to ensure consistency by always applying the same information to certain measurements.

5 Glossary

Property – a pair of values, one of which is the name of a field; the other represents the value of that field

Meta data – information about a test and all the nested references

Authentication system – enveloping system providing access rights to users for objects – i.e. whether users can create, read, update or delete them